**DSInventory-System Documentation (v1.0)**

The Devstonic teambrings you a powerful, flexible and scalable system called DSInventorySystem.

Imagine a world where inventory management is not a chore, but rather an exciting adventure. Where each item has its own unique history and meaning, and your players are immersed in an engaging gameplay experience. DSInventorySystem allows you to do just that.

At the heart of this system is the **Inventory Component**, a powerful tool that makes item management easy and intuitive. With it, players can easily add, remove, and organize their items, enjoying every minute of the game.

But what to do when the inventory is full? That's where the **Storage Component** comes in. This component allows you to create virtual storages, such as chests and boxes, where players can safely store their precious finds. Store your treasures and resources in convenient and accessible places, creating depth and realism in your game world.

And of course, what adventures without unique items? The **Item Component** allows you to create game artifacts with unique properties. The name, description, weight and other parameters make each item meaningful and special, which increases the excitement of exploring the world and collecting rare items.

But we don’t stop there. Imagine players creating their own unique items using the resources they find. The **Craft Component** opens up endless possibilities for crafting and creativity. Collect rare materials, combine them and create powerful weapons or valuable items that will help you in your adventures.

With the **Merchant Component**, you can create a living economic environment where players can buy and sell items. This component allows you to interact with NPCs and other players, creating an exciting and realistic trading system. Buy a rare artifact or sell surplus resources to earn more gold for your needs.

And don’t forget about looting! The **Loot Component** makes every battle meaningful and rewarding. Imagine players defeating powerful bosses and finding treasures that will help them become stronger and more successful in their adventures. This component adds excitement and interest to battles, encouraging players to explore the world and fight for rare and valuable trophies.

Why choose DSInventorySystem?

**- Intuitive interface:** Convenient and clear tools for inventory management, available to every player.

**- Powerful multiplayer support**: Ideal for online games with synchronization and interaction between players. (Multiplayer Inventory Component will be available in version 1.1!)

**- Flexibility and scalability:** Easily adapts to various game projects and developer needs.

**- Creative freedom**: Crafting and trading options add depth and strategy to the gameplay.

**- Exciting battles:** The loot system makes every battle meaningful and rewarding.

1. **Inventory Component Overview**

This component is responsible for managing an inventory system within a game.

It handles various operations related to adding, removing, and using items,

as well as managing inventory slots and weight constraints.

Key functionalities include:

- Adding and removing items

- Sorting and managing inventory slots

- Handling item usage and interaction with other systems such as crafting and storage

- Managing inventory weight and overload state

- Broadcasting events for inventory updates and interactions

This component can be used in any actor that requires an inventory system.

|  |  |
| --- | --- |
| **Variables** | |
| **Screenshot 2024-12-02 200329** | The array ([Inventory Slot Info structure](#InventorySlotInfo)) of inventory slots, storing information about each slot's items. This property is replicated across the network. |
| Screenshot 2024-12-02 200553 | A weak pointer to the storage component, used for interactions with storage.  This property is replicated across the network |
| Screenshot 2024-12-02 200625 | A weak pointer to the craft component, used for interactions with crafting.  This property is replicated across the network |
| Screenshot 2024-12-02 200636 | A weak pointer to the loot component, used for interactions with looting.  This property is replicated across the network |
| Screenshot 2024-12-02 200653Screenshot 2024-12-02 200653 | The quantity of slots available in the inventory.  This property is replicated across the network and can be increased using function [Increase Quantity Of Inventory Slots](#IncreaseQuantityOfInventorySlots). |
| Screenshot 2024-12-02 200711 | Flag indicating whether the weight system is used in the inventory. |
| Screenshot 2024-12-02 200714 | The maximum capacity the inventory can hold before it becomes overloaded.  This property is replicated across the network and can be increased or decreased using functions [Increase Overload Capacity](#IncreaseOverloadCapacity) and [Decrease Overload Capacity](#DecreaseOverloadCapacity). |
| Screenshot 2024-12-02 200838 | The current weight of items in the inventory.  This property is replicated across the network. |

|  |  |
| --- | --- |
| **Functions** | |
| Screenshot 2024-12-02 201718 | Finds an available inventory slot.    Found Index - The index of the found available slot.  Returns true if an available slot is found, otherwise false. |
| Screenshot 2024-12-02 201723 | Finds an available stack in the inventory that can accommodate the specified item.    Item to Search - The item to search for.  Found Index - The index of the found available stack.  Returns true if an available stack is found, otherwise false. |
| Screenshot 2024-12-02 201726 | Internally adds an item to the inventory.    Info ([AddItemToInventoryInfo structure](#AddItemToInventoryInfo)) - The information about the item to add.  Remainder - The remaining quantity that could not be added.  Returns true if the item is successfully added, otherwise false. |
| Screenshot 2024-12-02 201729 | Adds an item to the inventory.    Info ([AddItemToInventoryInfo structure](#AddItemToInventoryInfo)) - The information about the item to add.  Show Notify About Added Item - Whether to show a notification about the added item.  Remainder - The remaining quantity that could not be added.  Returns true if the item is successfully added, otherwise false. |
| Screenshot 2024-12-02 201736 | Adds an item to the inventory at the specified index.    Info ([AddItemToInventoryInfo structure](#AddItemToInventoryInfo)) - The information about the item to add.  Index - The index at which to add the item.  Returns true if the item is successfully added, otherwise false. |
| Screenshot 2024-12-02 201740 | Removes an item from the inventory by the specified index and quantity.    Index - The index of the item to remove.  Quantity - The quantity of the item to remove.  Returns true if the item is successfully removed, otherwise false. |
| Screenshot 2024-12-02 201744 | Throws an item from the inventory by the specified index and quantity.    Index - The index of the item to throw.  Quantity - The quantity of the item to throw.  Returns true if the item is successfully thrown, otherwise false. |
| Screenshot 2024-12-02 201749 | Increases the quantity of an item in the inventory by the specified index and quantity.    Index - The index of the item to increase.  Quantity - The quantity to increase the item by.  Returns true if the item's quantity is successfully increased, otherwise false. |
| Screenshot 2024-12-02 201753 | Swaps the items in the specified inventory slots.    Index From - The index of the first slot.  Index To - The index of the second slot.  Returns true if the items are successfully swapped, otherwise false. |
| Screenshot 2024-12-02 201757 | Splits the stack of items in the inventory at the specified index and quantity.    Index - The index of the stack to split.  Quantity - The quantity to split from the stack.  Returns true if the stack is successfully split, otherwise false. |
| Screenshot 2024-12-02 201801 | Moves an item in the inventory from one index to another.    Index From - The index of the item to move.  Index To - The index to move the item to.  Returns true if the item is successfully moved, otherwise false. |
| Screenshot 2024-12-02 201804 | Splits a stack of items in the inventory from one index to another with the specified quantity.    Index From - The index of the stack to split from.  Index To - The index to split the stack to.  Quantity - The quantity to split from the stack.  Returns true if the stack is successfully split, otherwise false. |
| Screenshot 2024-12-02 201807 | Uses an item in the inventory by the specified index.    Index - The index of the item to use. |
| Screenshot 2024-12-02 201810 | Increases the quantity of inventory slots.    New Quantity - The new total quantity of inventory slots. |
| Screenshot 2024-12-02 201813 | Increases the inventory weight by the specified quantity.    Quantity - The quantity to increase the inventory weight by. |
| Screenshot 2024-12-02 201816 | Decreases the inventory weight by the specified quantity.    Quantity - The quantity to decrease the inventory weight by. |
| Screenshot 2024-12-02 201820 | Increases the inventory weight by the weight of a specified item and quantity.    Item - The item whose weight will be used to increase the inventory weight.  Quantity - The quantity of the item to calculate the weight increase. |
| Screenshot 2024-12-02 201822 | Decreases the inventory weight by the weight of a specified item and quantity.    Item - The item whose weight will be used to decrease the inventory weight.  Quantity - The quantity of the item to calculate the weight decrease. |
| Screenshot 2024-12-02 201826 | Decreases the overload capacity of the inventory by the specified amount.    Capacity to Decrease - The amount to decrease the overload capacity by. |
| Screenshot 2024-12-02 201829 | Increases the overload capacity of the inventory by the specified amount.    Capacity to Increase - The amount to increase the overload capacity by. |
| Screenshot 2024-12-02 201832 | Sorts the inventory by item names.    Reversed - Whether to sort in reverse order. |
| Screenshot 2024-12-02 201835 | Sorts the inventory by item quantity.    Reversed - Whether to sort in reverse order. |
| Screenshot 2024-12-02 201838 | Sorts the inventory by item types based on a provided sort order.    Sort Order ([Item Type structure](#ItemType)) - The order in which to sort item types.  Reversed - Whether to sort in reverse order. |
| Screenshot 2024-12-02 201842 | Gets the valid inventory slots that contain items.    Returns an array of valid inventory slots. |
| Screenshot 2024-12-02 201846 | Moves an item from the loot inventory to the player's inventory by the specified indices.    Loot Index - The index of the item in the loot inventory.  Inventory Index - The index in the player's inventory to move the item to.  Returns true if the item is successfully moved, otherwise false. |
| Screenshot 2024-12-02 201849 | Moves an item from the storage to the player's inventory by the specified indices.    Storage Index - The index of the item in the storage.  Inventory Index - The index in the player's inventory to move the item to.  Returns true if the item is successfully moved, otherwise false. |
| Screenshot 2024-12-02 201853 | Swaps an item in the storage with an item in the player's inventory by the specified indices.    Storage Index - The index of the item in the storage.  Inventory Index - The index of the item in the player's inventory.  Returns true if the items are successfully swapped, otherwise false. |
| Screenshot 2024-12-02 201857 | Splits an item stack in the storage and moves a specified quantity to an inventory slot.    Storage Index - The index of the item stack in the storage.  Inventory Index - The index in the player's inventory to move the split quantity to.  Quantity - The quantity to split from the item stack.  Returns true if the item stack is successfully split and moved, otherwise false. |
| Screenshot 2024-12-02 201900 | Checks if the inventory is overloaded.    Returns true if the current weight exceeds the overload capacity, otherwise false. |
| Screenshot 2024-12-02 201908 | Gets all inventory slots.    Returns an array of all inventory slots ([Inventory Slot Info structure](#InventorySlotInfo)). |
| Screenshot 2024-12-02 201911 | Gets the current weight of the inventory.    Returns the current weight of the inventory. |
| Screenshot 2024-12-02 201914 | Checks if the specified inventory slot is available.    Index - The index of the inventory slot to check.  Returns true if the slot is available, otherwise false. |
| Screenshot 2024-12-02 201917 | Gets the items in the specified inventory slot.    Index - The index of the inventory slot.  Returns an array of items in the specified inventory slot. |
| Screenshot 2024-12-02 201920 | Gets the quantity of items in the specified inventory slot.    Index - The index of the inventory slot.  Returns the quantity of items in the specified inventory slot. |
| Screenshot 2024-12-02 201924 | Gets the first item in the specified inventory slot.    Index - The index of the inventory slot.  Returns the first item in the specified inventory slot, or nullptr if no items are present. |
| Screenshot 2024-12-02 201927 | Checks if the inventory is full.    Returns true if the inventory is full, otherwise false. |
| Screenshot 2024-12-02 201930 | Sets the inventory slot at the specified index with the provided item information.    Index - The index of the inventory slot to set.  Info ([Add Item To Inventory Info structure](#AddItemToInventoryInfo)) - The information about the item to set in the inventory slot. |
| Screenshot 2024-12-02 201933 | Clears a specified quantity of items from the inventory slot at the given index.    Index - The index of the inventory slot to clear items from.  Quantity - The quantity of items to clear from the inventory slot. |
| Screenshot 2024-12-02 201936 | Creates an item object of the specified class.    Item to Create - The class of the item to create.  Returns a pointer to the newly created item object. |
| Screenshot 2024-12-02 201938 | Finds the index of the specified item in the inventory.    Item - The item to find.  Returns the index of the item if found, otherwise INDEX\_NONE. |

|  |  |
| --- | --- |
| **Events** | |
| Screenshot 2024-12-02 201941 | Event triggered when the inventory is full (if you configure it, since in the code this event is not called).  This function is a BlueprintNativeEvent. |
| Screenshot 2024-12-02 201945 | Event triggered when it is necessary to throw away an item. |
| Screenshot 2024-12-02 201949 | Event triggered when an inventory slot needs to be updated by index. |
| Screenshot 2024-12-02 201952 | Event triggered to show a notification about an added item. |
| Screenshot 2024-12-02 201955 | Event triggered when an item is added to the inventory |
| Screenshot 2024-12-02 202000 | Event triggered when the inventory begins to be overloaded. |
| Screenshot 2024-12-02 202003 | Event triggered when the inventory is no longer overloaded. |
| Screenshot 2024-12-02 202006 | Event triggered when an item is used. |
| Screenshot 2024-12-02 202010 | Event triggered when an item is removed from the inventory. |
| Screenshot 2024-12-02 202013 | Event triggered when inventory slots need to be generated. |

1. **Storage Component Overview**

This component is responsible for managing a storage system within a game.

It handles various operations related to adding, removing, and organizing items within storage slots.

The storage component can interact with the player's inventory to facilitate item transfers and management.

Key functionalities include:

- Initialization of storage slots and generation of starter content

- Adding and removing items from storage

- Sorting and managing storage slots

- Interacting with the player's inventory for item transfers

- Broadcasting events for storage updates and interactions

This component can be used in any actor that requires a storage system.

|  |  |
| --- | --- |
| **Variables** | |
| Screenshot 2024-12-02 203417 | The data table containing initial content information for the storage. |
| Screenshot 2024-12-02 203429 | The array ([Inventory Slot Info structure](#InventorySlotInfo)) of storage slots containing information about each slot's items.  This property is replicated across the network. |
| Screenshot 2024-12-02 203431 | Weak pointer to the inventory component for interacting with the player's inventory.  This property is replicated across the network. |
| Screenshot 2024-12-02 203433 | The quantity of slots available in the storage. |

|  |  |
| --- | --- |
| **Functions** | |
| Screenshot 2024-12-02 203914 | Initializes the storage component with the specified quantity of slots.    Quantity Of Slots Param - The quantity of slots to initialize. |
| Screenshot 2024-12-02 203917 | Generates the starter content for the storage component. |
| Screenshot 2024-12-02 203920 | Constructs the storage data using the specified inventory component reference.    Inventory Component Ref - A reference to the inventory component. |
| Screenshot 2024-12-02 203923 | Destructs the storage data, clearing all associated data. |
| Screenshot 2024-12-02 203927 | Finds an available storage slot.    Found Index - The index of the found available slot.  Returns true if an available slot is found, otherwise false. |
| Screenshot 2024-12-02 203930 | Finds an available stack in the storage that can accommodate the specified item.    Item to Search - The item to search for.  Found Index - The index of the found available stack.  Returns true if an available stack is found, otherwise false. |
| Screenshot 2024-12-02 203934 | Adds an item to the storage.    Info ([Add Item To Inventory Info structure](#AddItemToInventoryInfo)) - Information about the item to add to the storage.  Remainder - The remaining quantity that could not be added.  Returns true if the item is successfully added, otherwise false. |
| Screenshot 2024-12-02 203938 | Adds an item to the storage at the specified index.    Info ([Add Item To Inventory Info structure](#AddItemToInventoryInfo)) - Information about the item to add to the storage.  Index - The index at which to add the item.  Returns true if the item is successfully added, otherwise false. |
| Screenshot 2024-12-02 203941 | Removes an item from the storage by the specified index and quantity.    Index - The index of the item to remove.  Quantity - The quantity of the item to remove.  Returns true if the item is successfully removed, otherwise false. |
| Screenshot 2024-12-02 203944 | Increases the quantity of an item in the storage by the specified index and quantity.    Index - The index of the item to increase.  Quantity - The quantity to increase the item by.  Returns true if the item's quantity is successfully increased, otherwise false. |
| Screenshot 2024-12-02 203947 | Swaps the items in the specified storage slots.    Index From - The index of the first slot.  Index To - The index of the second slot.  Returns true if the items are successfully swapped, otherwise false. |
| Screenshot 2024-12-02 203951 | Splits the stack of items in the storage at the specified index and quantity.    Index - The index of the stack to split.  Quantity - The quantity to split from the stack.  Returns true if the stack is successfully split, otherwise false. |
| Screenshot 2024-12-02 203953 | Moves an item in the storage from one index to another.    Index From - The index of the item to move.  Index To - The index to move the item to.  Returns true if the item is successfully moved, otherwise false. |
| Screenshot 2024-12-02 203956 | Splits a stack of items in the storage from one index to another with the specified quantity.    Index From - The index of the stack to split from.  Index To - The index to split the stack to.  Quantity - The quantity to split from the stack.  Returns true if the stack is successfully split, otherwise false. |
| Screenshot 2024-12-02 203959 | Sorts the storage by item names.    Reversed - Whether to sort in reverse order. |
| Screenshot 2024-12-02 204002 | Sorts the storage by item quantity.    Reversed - Whether to sort in reverse order. |
| Screenshot 2024-12-02 204006 | Sorts the storage by item types based on a provided sort order.    Sort Order ([Item Type structure](#ItemType)) - The order in which to sort item types.  Reversed - Whether to sort in reverse order. |
| Screenshot 2024-12-02 204009 | Gets the valid storage slots that contain items.    Returns an array of valid storage slots ([Inventory Slot info structure](#InventorySlotInfo)). |
| Screenshot 2024-12-02 204011 | Moves an item from the inventory to the storage by the specified indices.    Storage Index - The index of the item in the storage.  Inventory Index - The index in the inventory to move the item from.  Returns true if the item is successfully moved, otherwise false. |
| Screenshot 2024-12-02 204015 | Swaps an item in the inventory with an item in the storage by the specified indices.    Storage Index - The index of the item in the storage.  Inventory Index - The index of the item in the inventory.  Returns true if the items are successfully swapped, otherwise false. |
| Screenshot 2024-12-02 204018 | Splits an item stack in the inventory and moves a specified quantity to a storage slot.    Storage Index - The index of the item stack in the storage.  Inventory Index - The index in the inventory to move the split quantity from.  Quantity - The quantity to split from the item stack.  Returns true if the item stack is successfully split and moved, otherwise false. |
| Screenshot 2024-12-02 204021 | Gets all storage slots.    Returns an array of all storage slots ([Inventory Slot Info structure](#InventorySlotInfo)). |
| Screenshot 2024-12-02 204024 | Checks if the specified storage slot is available.    Index - The index of the storage slot to check.  Returns true if the slot is available, otherwise false. |
| Screenshot 2024-12-02 204027 | Gets the items in the specified storage slot.    Index - The index of the storage slot.  Returns an array of items in the specified storage slot. |
| Screenshot 2024-12-02 204812 | Gets the quantity of items in the specified storage slot.    Index - The index of the storage slot.  Returns the quantity of items in the specified storage slot. |
| Screenshot 2024-12-02 204030 | Gets the first item in the specified storage slot.    Index - The index of the storage slot.  Returns the first item in the specified storage slot, or nullptr if no items are present. |
| Screenshot 2024-12-02 204034 | Sets the storage slot at the specified index with the provided item information.    Index - The index of the storage slot to set.  Info ([Add Item To Inventory Info structure](#AddItemToInventoryInfo)) - Information about the item to set in the storage slot. |
| Screenshot 2024-12-02 204036 | Clears a specified quantity of items from the storage slot at the given index.    Index - The index of the storage slot to clear items from.  Quantity - The quantity of items to clear from the storage slot. |
| Screenshot 2024-12-02 204039 | Creates an item object of the specified class.    Item to Create - The class of the item to create.  Returns a pointer to the newly created item object. |

|  |  |
| --- | --- |
| **Events** | |
| Screenshot 2024-12-02 204042 | Event triggered when a storage slot need to be updated by index. |
| Screenshot 2024-12-02 204046 | Event triggered when an item is added to the storage. |
| Screenshot 2024-12-02 204050 | Event triggered when an item is removed from the storage. |

1. **Craft Component Overview**

This component is responsible for managing the crafting system within a game.

The CraftComponent interacts with the player's inventory and broadcasts updates to the crafting menu.

Key functionalities include:

- Initialization and generation of crafting content

- Interacting with the player's inventory for crafting operations

- Broadcasting events for crafting menu updates

This component can be used in any actor that requires a crafting system.

|  |  |
| --- | --- |
| **Variables** | |
| Screenshot 2024-12-02 205304 | An array of item subclasses that can be crafted.  This property is replicated across the network |
| Screenshot 2024-12-02 205306 | A reference to the data table containing crafting content information. |
| Screenshot 2024-12-02 205308 | A weak pointer to the inventory component, used for interactions with the player's inventory.  This property is replicated across the network. |

|  |  |
| --- | --- |
| **Functions** | |
| Screenshot 2024-12-02 205311 | Generates the crafting content for the crafting system.  This function is called during initialization to populate the crafting system with available recipes and items. |
| Screenshot 2024-12-02 205315 | Constructs the crafting data using the specified inventory component reference.  Inventory Component Ref - A reference to the inventory component to use for constructing crafting data. |
| Screenshot 2024-12-02 205318 | Destructs the crafting data, clearing any associated data and resources.  This function is called during shutdown or when resetting the crafting system. |
| Screenshot 2024-12-02 205322 | Uses a specified quantity of the item for crafting.  Item - The class of the item to use.  Quantity - The quantity of the item to use for crafting.  Returns true if the item was successfully used, otherwise false. |
| Screenshot 2024-12-02 205325 | Gets the total quantity of the specified item from the inventory.  Item - The class of the item to search for.  Indexes Of Items - An array to be filled with the indexes of the found items in the inventory.  Returns The total quantity of the specified item found in the inventory. |

|  |  |
| --- | --- |
| **Events** | |
| Screenshot 2024-12-02 205331 | An event that broadcasts when the crafting menu is updated.  This event can be used to notify the UI or other systems when the crafting menu changes. |

1. **Merchant Component Overview**

This component is responsible for managing the merchant system within a game.

The MerchantComponent interacts with the player's inventory and broadcasts updates to the merchant menu.

Key functionalities include:

- Initialization and generation of merchant content

- Managing merchant offers and item quantities

- Interacting with the player's inventory for trading operations

- Broadcasting events for merchant menu updates

This component can be used in any actor that requires a trading or merchant system.

|  |  |
| --- | --- |
| **Variables** | |
| Screenshot 2024-12-02 205916 | The data table containing the merchant's content information. |
| Screenshot 2024-12-02 205919 | The array ([Merchant Offer Info structure](#MerchantOfferInfo)) of merchant offers.  This property is replicated across the network. |
| Screenshot 2024-12-02 205921 | A weak pointer to the inventory component, used for interactions with the player's inventory.  This property is replicated across the network. |

|  |  |
| --- | --- |
| **Functions** | |
| Screenshot 2024-12-02 205925 | Generates the merchant content for the merchant system.  This function is called during initialization to populate the merchant with available items. |
| Screenshot 2024-12-02 205928 | Constructs the merchant data using the specified inventory component reference.  Inventory Component Ref - A reference to the inventory component to use for constructing merchant data. |
| Screenshot 2024-12-02 205931 | Destructs the merchant data, clearing any associated data and resources.  This function is called during shutdown or when resetting the merchant system. |
| Screenshot 2024-12-02 205935 | Gets the total quantity of the specified item from the inventory.  Item - The class of the item to search for.  Indexes Of Items - An array to be filled with the indexes of the found items in the inventory.  Returns the total quantity of the specified item found in the inventory. |
| Screenshot 2024-12-02 205938 | Removes a merchant offer by the specified item.  Item - The class of the item to remove from the merchant's offers.  Returns true if the offer was successfully removed, otherwise false. |
| Screenshot 2024-12-02 205941 | Increases the quantity of a merchant offer by the specified value.  Item - The class of the item to increase the quantity of.  Value - The amount to increase the offer quantity by. |
| Screenshot 2024-12-02 205944 | Decreases the quantity of a merchant offer by the specified value.  Item - The class of the item to decrease the quantity of.  Value - The amount to decrease the offer quantity by. |

|  |  |
| --- | --- |
| **Events** | |
| Screenshot 2024-12-02 205950 | This event can be used to notify the UI or other systems when the merchant menu changes. |

1. **Loot Component Overview**

This component manages the loot inventory system within a game.

It handles the initialization, addition, removal, and querying of loot items,

as well as interacting with the player's inventory and other game systems.

Key functionalities include:

- Initializing loot data from an inventory component or a data table

- Adding and removing items to and from the loot inventory

- Querying and managing loot slots and their contents

- Broadcasting events related to loot updates and interactions

This component can be used in any actor that requires a loot inventory system.

|  |  |
| --- | --- |
| **Variables** | |
| **Screenshot 2024-12-02 210725** | The data table containing information about the loot content. |
| **Screenshot 2024-12-02 210732** | The array ([Inventory Slot Info structure](#InventorySlotInfo)) of loot slots containing information about each slot's items.  This property is replicated across the network. |
| **Screenshot 2024-12-02 211135** | Weak pointer to the inventory component for interacting with the player's inventory.  This property is replicated across the network. |

|  |  |
| --- | --- |
| **Functions** | |
| Screenshot 2024-12-02 210735 | Initializes the loot component using the specified inventory component reference.    Inventory Component Ref - A reference to the inventory component. |
| Screenshot 2024-12-02 210738 | Initializes the loot component using the data from the data table. |
| Screenshot 2024-12-02 210741 | Constructs the loot data using the specified inventory component reference.    Inventory Component Ref - A reference to the inventory component. |
| Screenshot 2024-12-02 210745 | Destructs the loot data, clearing all associated data. |
| Screenshot 2024-12-02 210748 | Adds an item to the loot inventory.    Info ([Inventory Slot info structure](#InventorySlotInfo)) - Information about the item to add to the loot inventory.  Returns true if the item is successfully added, otherwise false. |
| Screenshot 2024-12-02 210753 | Removes an item from the loot inventory by the specified index.    Index - The index of the item to remove.  Returns true if the item is successfully removed, otherwise false. |
| Screenshot 2024-12-02 210757 | Removes an item from the loot inventory by the specified slot information.    Info ([Inventory Slot info structure](#InventorySlotInfo)) - The slot information of the item to remove.  Returns true if the item is successfully removed, otherwise false. |
| Screenshot 2024-12-02 210801 | Gets all loot slots.    Returns an array ([Inventory Slot Info structure](#InventorySlotInfo)) of all loot slots. |
| Screenshot 2024-12-02 210804 | Checks if the specified loot slot is available.    Index - The index of the loot slot to check.  Returns true if the slot is available, otherwise false. |
| Screenshot 2024-12-02 210808 | Gets the items in the specified loot slot.    Index - The index of the loot slot.  Returns an array of items in the specified loot slot. |
| Screenshot 2024-12-02 210812 | Gets the quantity of items in the specified loot slot.    Index - The index of the loot slot.  Returns the quantity of items in the specified loot slot. |
| Screenshot 2024-12-02 210815 | Gets the first item in the specified loot slot.    Index - The index of the loot slot.  Returns the first item in the specified loot slot, or nullptr if no items are present. |
| Screenshot 2024-12-02 210819 | Checks if the loot inventory is empty.    Returns true if the loot inventory is empty, otherwise false. |
| Screenshot 2024-12-02 210822 | Creates an item object of the specified class.    Item to Create - The class of the item to create.  Returns a pointer to the newly created item object. |

|  |  |
| --- | --- |
| **Events** | |
| Screenshot 2024-12-02 210827 | Event triggered when loot when slots need to be generated |
| Screenshot 2024-12-02 210830 | Event triggered when an item is removed from the loot. |
| Screenshot 2024-12-02 210834 | Event triggered when an item is added to the loot. |

1. **Item Component Overview**

This class represents an item within the game. It manages properties related to the item,

such as its name, description, icon, usage, stackability, type, durability, weight,

and associated interaction, crafting, and purchase data.

Key functionalities include:

- Initialization of item properties from data structures

- Management of item durability and usage

- Interaction with the player's inventory and other game systems

- Broadcasting events for item state changes, such as durability reaching zero

This class can be used as a base class for specific items within the game, allowing for

customization and extension of item behavior.

|  |  |
| --- | --- |
| **Variables** | |
| Screenshot 2024-12-02 211742 | The current durability of the item. |
| Screenshot 2024-12-02 211744 | The data ([Item Data structure](#ItemData)) associated with the item, including its properties like name, description, icon, and more.  This property is replicated across the network. |

|  |  |
| --- | --- |
| **Functions** | |
| Screenshot 2024-12-02 211747 | Initializes the item with the provided data and current durability.  In Data ([Item Data structure](#ItemData)) - The data to initialize the item with.  In Current Durability - The current durability of the item. |
| Screenshot 2024-12-02 211750 | Initializes the item with the data from another item object.  Item - The item object to initialize the item with. |
| Screenshot 2024-12-02 211753 | Uses the item, performing any associated actions such as applying effects or consuming resources.  Inventory - The inventory component that contains the item.  Index - The index of the item in the inventory. |
| Screenshot 2024-12-02 211756 | Increases the current durability of the item by the specified value.  Value - The amount to increase the item's durability by. |
| Screenshot 2024-12-02 211759 | Decreases the current durability of the item by the specified value.  Value - The amount to decrease the item's durability by. |
| Screenshot 2024-12-02 211802 | Gets the current durability of the item.  Returns the current durability of the item. |
| Screenshot 2024-12-02 211806 | Gets the data associated with the item.  Returns the data ([Item Data structure](#ItemData)) associated with the item. |
| Screenshot 2024-12-02 211809 | Gets the inventory component that contains the item.  Returns a pointer to the inventory component. |

|  |  |
| --- | --- |
| **Events** | |
| Screenshot 2024-12-02 211812 | Event triggered when the item is broken.  This function can be implemented in Blueprints to define custom behavior when the item breaks. |
| Screenshot 2024-12-02 211817 | Delegate for when the item's durability reaches zero. |

1. **Multiplayer Inventory Component Overview**

**Multiplayer Inventory Component will be available in version 1.1.**

1. **Inventory System Game Instance Overview**

This class represents the game instance, which is responsible for managing the overall state of the game.

It handles operations related to saving and loading game data, as well as managing inventory settings.

The GameInstance is used to persist data across different levels and sessions, ensuring a seamless player experience.

Key functionalities include:

- Saving and loading game state

- Managing inventory settings

- Interacting with save game objects

This class can be extended to include additional game-wide functionalities as needed.

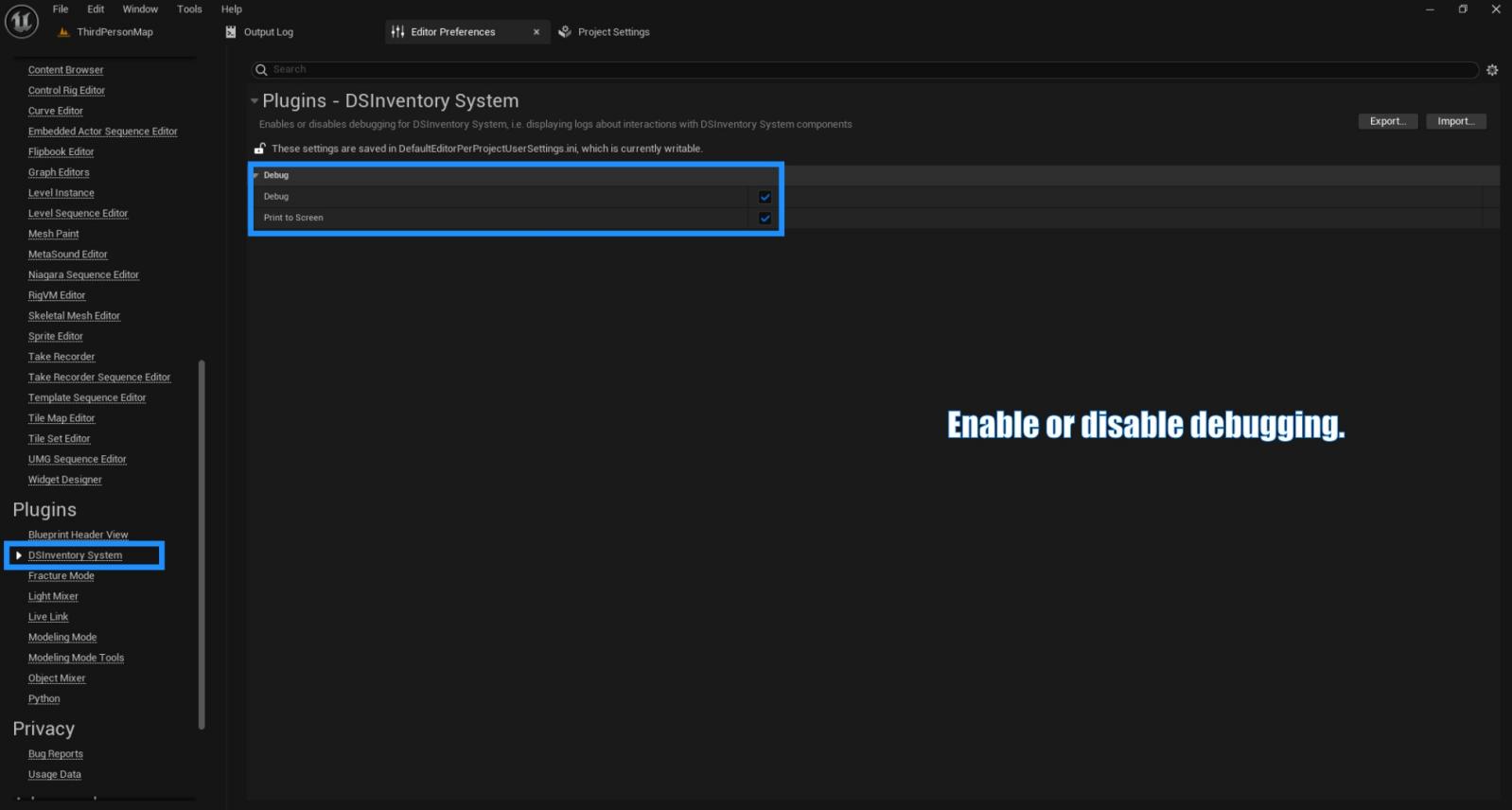
|  |  |
| --- | --- |
| **Variables** | |
| Screenshot 2024-12-03 191112 | The name of the save game slot.  This property is read and writable in Blueprints and can be edited anywhere in the editor. |
| Screenshot 2024-12-03 191115 | Indicates whether a saved game should be loaded.  This property is read and writable in Blueprints and can be edited anywhere in the editor. |
| Screenshot 2024-12-03 191118 | The maximum stack size for inventory items. |
| Screenshot 2024-12-03 191121 | The save game object used for saving and loading game data. |

|  |  |
| --- | --- |
| **Events** | |
| Screenshot 2024-12-03 191124 | Saves the game state.  This function can be called in Blueprints and can be implemented in Blueprints. |
| Screenshot 2024-12-03 191128 | Loads the game state.  This function can be called in Blueprints and can be implemented in Blueprints. |

1. **All Structures Overview**

|  |  |
| --- | --- |
| **Item Data -**  This structure holds general item data, including the item's name, description,  icon, usability, stackability, type, durability, weight, and associated interaction,  crafting, and purchase data. | **Screenshot 2024-12-02 212215** |
| **Item Purchase Data -** This structure holds data for item purchases, including the value of the item. | **Screenshot 2024-12-02 212340** |
| **Interaction Data -**  This structure holds data for interaction, including the actor for interaction,  the mesh for interaction, the material to apply, and the scale of the interaction. | **Screenshot 2024-12-02 212332** |
| **Item Craft Data -**  This structure holds data related to item crafting, such as whether the item  can be crafted, the items required for crafting, and the crafting delay. | **Screenshot 2024-12-02 212327** |
| **Craftable Item Class info -** This structure holds information about a craftable item class,  including the item class itself and the quantity required for crafting. | **Screenshot 2024-12-02 212415** |
| **Item Type -** Enumeration for different types of items in the game. Each item type has a specific purpose or role. | **Screenshot 2024-12-03 103932** |
| **Item Rarity -**  Enumeration for different rarity levels of items in the game. | **Screenshot 2024-12-03 103936** |
| **Inventory Slot Info -**  A structure that represents information about an inventory slot.    This structure is used to store items within an inventory slot. | **Screenshot 2024-12-02 213100** |
| **Add Item To Inventory Info** - A structure that represents information for adding an item to the inventory.    This structure is used to specify the item and the quantity to be added to the inventory. | **Screenshot 2024-12-02 213123** |
| **Storage Content Data Row -**  This structure represents the content of storage and inherits from FTableRowBase.  It is used to define the items that can be stored, along with their minimum and maximum quantities,  and the chance of their appearance. | **Screenshot 2024-12-02 213241** |
| **Craft Content Data Row** - This structure is used to define the content of a crafting data row.  It inherits from FTableRowBase, making it suitable for use in data tables. | **Screenshot 2024-12-02 213335** |
| **Merchant Content Data Row** - This structure represents the data row for merchant content in a data table.  It inherits from FTableRowBase, making it suitable for use in data tables. | **Screenshot 2024-12-02 213454** |
| **Merchant Offer Info** - This structure holds information about a merchant's offer, including the item being offered,  whether the quantity is limited, and the quantity of the item. | **Screenshot 2024-12-02 213437** |
| **Loot Inventory Info** - This structure represents the content of the loot inventory and inherits from [FTableRowBase](https://dev.epicgames.com/documentation/en-us/unreal-engine/data-driven-gameplay-elements-in-unreal-engine).  It is used to define the items that can be found as loot, along with their minimum and maximum quantities. | **Screenshot 2024-12-02 213607** |

**How to enable debagging**

****