Too Many Eyes / Inventory System

System Design Document

# Changes

## V 1.0

Editor: Perrin Peterson

11/06/2022

* Document Created, and initial version

## V 1.1

Editor: Perrin Peterson

11/10/2022

* Added introduction.
* Changed document name.

## V 1.2

Editor: Perrin Peterson

11/11/2022

* Updated after bug fixes.

## V 2.0

Editor: Perrin Peterson  
12/02/2022

* Major overhaul and rework.

# Introduction

This document details the design and purpose of a locomotion system designed for Too Many Eyes. Designed for the Blight Brew Game, the document will cover;

* Usage for the designers.
* UML for the Programmers.

# Design Goals

The end goal for this system is to give the player an inventory that’s manageable, and fluid enough to give the player plenty of information at a glance. The inventory should be organizable, and customizable.

# Behaviour

This system is mainly focused on supplying the player with a container that they can hold items they:

* Find in the world
* Craft
* Receive from any sources

For balancing purposes, the system is smart enough to perform basic organisation tasks. The system will automatically sort items into containers and will not allow more than the maximum stack size. Many of the intelligence of this system will be used by the UI system, but the Inventory system is built with this in mind.

In version 2.0 of this system, the overhaul gave us some major optimisation changes. Items no longer exist in the world while held by the player, or any other inventory. This system now also encompasses *Storage*, which can be seen as a child of this system.

# High Level Design

Graphical user interface

Description automatically generated with medium confidence

* Player system - the system is our core system and is what the player interacts with to use all our other systems.
* Locomotion system - an advanced movement system we’re using to give us more control over the players movement. This gives us variables we can control, such as stamina, and additional functionality, such as climbing and vaulting.
* Potion system – a system meant to be easy to use for designers. The system allows for the designers as much freedom as possible, while giving the coders a minimal amount of updating to do. This is the main system the player will use to complete puzzles and generate income.
* Inventory system – a system meant to be easy to use for designers. The system allows for the designers as much freedom as possible, while giving the coders a minimal amount of updating to do. The system works as a container to hold items that the player collects. This includes the players backpack, as well as storage containers around the world.
* Resource system – a system meant to be the first of the steps for the player to generate income. This system gives the player ways to harvest materials and shows off behavioural logic for when the system is used.
* Crafting System – the system in between the Resource system, and the Shop/Potion systems, in terms of actual gameplay. This systems job is to refine the resources the player gathers into other items or potions for use in one of the other systems.
* Shop system (Not yet implemented) – a system to generate income, and allow for the player to improve. The system interfaces with AI, the player, and Storage objects, to give the player a way of selling items to NPC’s. This can have numerous effects on the AI and is the primary source of income, a necessary resource for improving the players arsenal, and serves to break the monotony of just grinding for resources.
* AI System - The system dealing with the NPC’s in the world, including fauna. The system interfaces with the shop system, allowing for NPC’s to be customers, as well as gives the AI their logic for movement, interaction, and anything else.

# Mid Level View

Graphical user interface

Description automatically generated

*For information about the ABBPlayerCharacter, and ABBCharacterBase, please visit the Player MDD.*

* UBBInventoryCompont is the inventory manager and hold many of the functions to communicate between the inventory slots in holds, and the inventory slots in other objects in the game.
* UBBInventorySlotNew holds onto the data about the item that it stores, as well as has functionality to modify its own data.
* ABBBaseItem is the Base class for all the items in the game. It uses an Interface class that gives it focus text and other functionality. It holds onto data to be transferred to the user, to give it identity, such as a name, description, and an Icon.
* ABBStorageContaner is the base for all Storage actors in the game, like chests and shelves.
* ABBDisplayStand is a storage container for holding and displaying 1 stack of items. It contains separate StaticMeshComponent that displays the mesh of the item it’s holding in its Inventory Slot, and a box that allows the mesh to be scaled down to a reasonable size.

# Logical View

Text

Description automatically generated

Text

Description automatically generated

A picture containing table

Description automatically generated

Table

Description automatically generated with medium confidence

Text

Description automatically generated

# Process View

## Displaying an item on the display stand

Diagram, text

Description automatically generated

1. If there is an Item in the storage slot;
2. We set the internal mesh’s relative scale to 1, 1, 1.
3. We get the items mesh,
4. Then we set it to the internal mesh.
5. We get the items texture,
6. Then we set it to the internal mesh.
7. We enter a While loop to try and shrink the mesh down to the size of the BoxComponent. Maximum 3 tries.

# Use Case View

## Changing inventory settings.

Currently, the only setting that can be effected for the inventory is the Max Size, which changes how many Inventory Slots the player has. This is found in the property panel in the blueprint editor of the player.

Graphical user interface

Description automatically generated

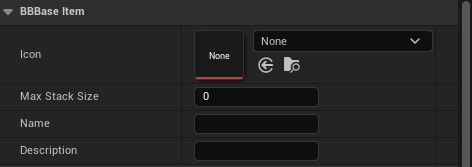
## Making custom items

Making a custom item is very simple. Make a BP based on the BaseItem class, and change the settings to suit your needs.

## Changing item settings

Items can be modified in the following ways:

* Icon
* Name
* Description
* MaxStackSize

These settings can all be modified in the BP editor. 

As well, the Mesh component is editable, for custom meshes.