

Hybrid BASS/Kyrandia template for AGS

by brewton

Description

This interface template combines my favourite interface features from the BASS template and The Legend of Kyrandia.

Features

- Single click on inventory items to pick up or pop out of slots.
- Retains useful right-click and left-click handling from BASS template.
- Custom room text displaying in the status bar.
- Kyrandia style - Player can carry items between rooms without having to slot them in inventory.
- Kyrandia 2 style inventory scrolling (non-looping)
- Custom cursors for displaying room exits on mouseover.
- Dummy item generation.

Walkable areas

Since dropped inventory items are actually characters, they can get stuck behind walk-behinds. Walkable Area ID # 2 is used by the `drop_item()` script to prevent items being placed there. To take advantage of this feature, on your walkable area masks use pallete index #2 for large areas that intersect the walk behind mask. See below for example.

		
Visible	Walkable Area Mask	Walk Behind Mask

Scripts

- TwoClickHandler.asc
A heavily modified version of the TwoClickHandler from the BASS stub. The hideable controls have been removed and many functions are delegated to KyrHandler.
- KyrHandler.asc
Handles most of the interface parts not handled by TwoClickHandler.
- DummyItems.asc
Handles all parts of generating and destroying dummy characters from inventory items.

Functions

KyrHandler.asc

- `get_exit_type()`
Returns a directional exit arrow based on the value of ExitType in the active hotspot.
- `drop_held_item()`
Drops the item held in the cursor and summons a dummy character.
- `pick_up_item(character)`
Adds the inventory item to the cursor slot and destroys the dummy character.
- `add_item(item)`
For instances where we just want to put an item in the cursor, like when an item is changed.
- `dummy_item_interaction()`
Returns a boolean value to the unhandled event script. If the character is not a dummy it will default to unhandled character interaction.

- *get_item_index(item)*
Finds which inventory slot an item is in.
- *return_blank()*
Populates a blank inventory slot based on which inventory slot has been used.
- *take_item_from_slot(item)*
Puts an inventory item from a slot into the cursor
- *drop_item_in_slot(item)*
Puts an inventory item from the cursor into a slot
- *switch_item_in_slot(item)*
Switches items between the cursor and a slot.
- *inv_scroll_up()*
inv_scroll_down()
Scrolls the inventory up or down.
- *player_self()*
Handles when the player clicks on themselves.

DummyItems.asc

- *init_dummies()*
Initializes array of dummy characters.
- *get_next_dummy()*
Searches through the dummyItem array for an available dummy character.
If there are no dummy characters left, the player will say “Dummy maximum reached!”
If this happens, increase the amount of dummy characters.
- *create_dummy(item)*
Creates a dummy character at the mouse position based on an inventory item
- *destroy_dummy(character)*
Resets the dummy character and restores it to the pool of available characters.

Custom variables

- *int* *invSlots* default = 15
This is the amount of inventory slots you're using.
This is used to determine the maximum amount of slots you have by indexing the inventory window.
If you want more slots, make sure you increase this number and create more blank items in the range 1-invSlots.
Note that you actually have invSlots + 1 since the inventory index starts at 0. The end index is used for the inventory item held in the cursor.
- *int* *invRows* default = 3
The maximum rows in the inventory bar. Used for inventory scrolling. Prevents showing the cursor item in the inventory bar.
- *int* *offset* default = 8
This is an offset about half the cursor height - so items don't jump all around the screen when swapping them.
- *int* *dummy_count* (int) default = 30
This is the maximum amount of dummy items that can be dropped throughout the game at a time.
If you are experiencing a "Dummy Maximum Reached!" state, you can increase this number, but be sure to
 - change it in the header file
 - modify the array total
 - and create additional dummy characters.

Custom Properties

There are a few custom properties that need to be set up to get the most out of this template.

Name	Description	Type	Default Value	Applies to
RoomName	Displays in the status bar. Useful if you want custom text to show up but still want a friendly name in Description.	Text	Empty Room	Rooms
GameArea	Which region of the game the player is in. Used for displaying different text from the character if they're going between regions. If set to 0, player will say "Where	Number	0	Rooms

	the heck am I?”			
ExitType	What kind of directional arrow to show over a hotspot. Used for showing room exits or entrances. 0 = Normal 1 = Up 2 = Right 3 = Down 4 = Left 5 = NoExit	Number	0	Hotspots
InvFrame	Inventory item’s sprite number in vInvItems Allows a single view to hold all inventory item sprites.	Number	0	Inventory Items