

Drag and Drop System



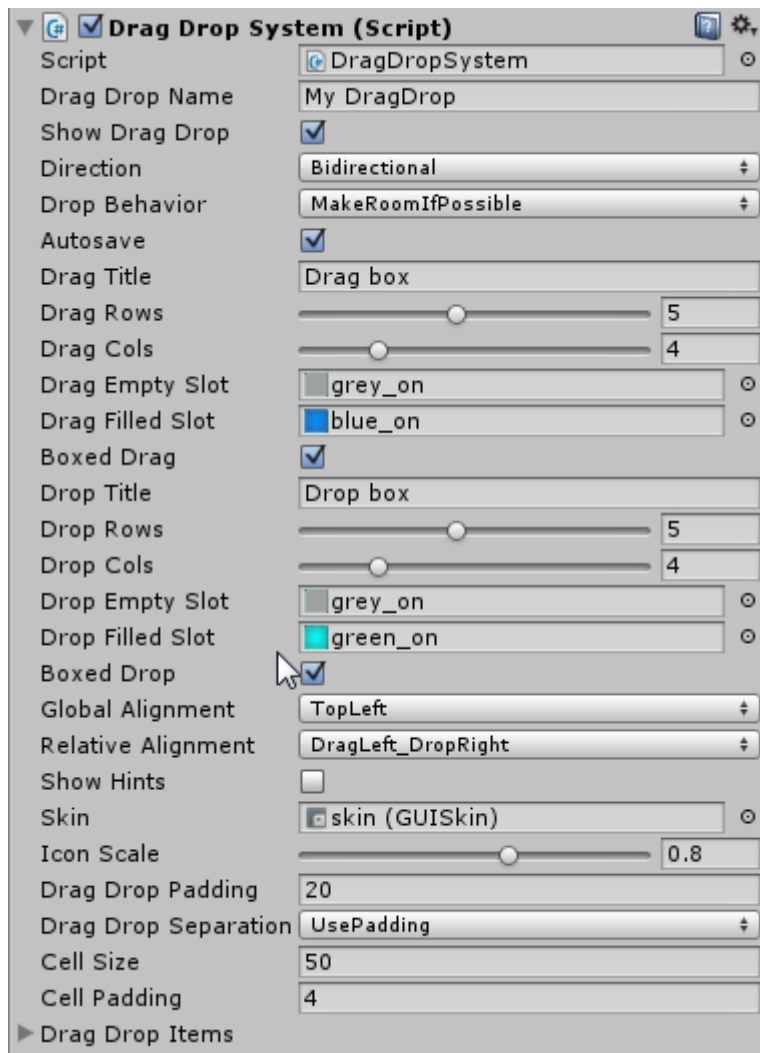
Create a Drag and Drop system for your game is really easy.

- 1) Drop the script on any object in your scene.
- 2) Set the scrip properties: Rows, columns, behavior...
- 3) Add as many items as you wish to the 'Drag and Drop Items' array
- 4) Run!

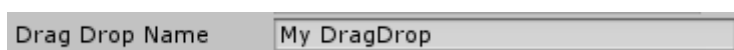
Modify the look and feel of any element. You can set the number of rows and columns for drag and drop boxes, cell background and size, box alignment (left-right or top-down), text color, custom skin, icon size, drag behavior, etc... The script can also add persistence to the drag and drop system so the inner state will be preserved during different executions.

There is a Trigger option to which notify external script when a drop happens, and populate some public variables, origin slot, destination slot, item index, error code, etc... See **Trigger** section in this manual.

Inspector options



Drag Drop Name



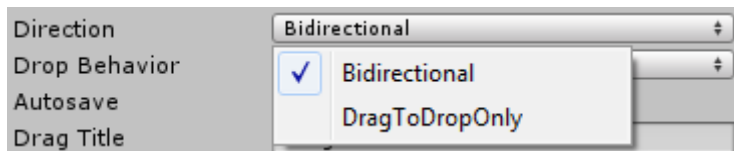
The name you wish to use for this instance of the object. This name will be used to load and save the data to persistent files if you set the Autosave to true.

Show Drag Drop



A public flag to switch on or off the whole Drag and Drop system. Set to false (unchecked) when you wish to hide the Drag and Drop.

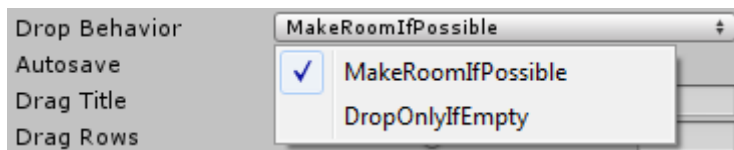
Direction



-Bidirectional: You can drag and drop items from any of the two boxes. That is, you can drag from the 'drag box' and drop in the 'drop box' and vice-versa.

-DragToDropOnly. You can only drag items from the 'drag box' and drop it on the 'drop box'.

Drop Behavior



MakeRoomIfPossible: If you drop the item over a filled slot, the system will try to move the item to an empty slot.

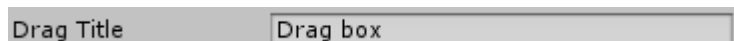
DropOnlyIfEmpty: You can only drop items over empty slots. If you drop the item over a filled slot the drop will be cancelled.

Autosave



If set to on, the drag and drop will have persistence. The system will automatically save the drag and drop states when any changes happens and will load when the application starts.

Drag Title



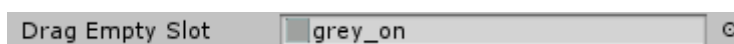
This option sets the text to display in the top of the Drag box. If set to blank no title will be displayed in the box. This property has a similar one for the Drop box 'Drop Title'.

Drag Rows – Drag Cols



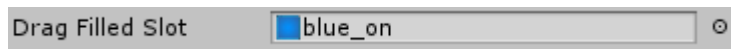
This option sets the number of rows and columns of the Drag Box. Drag Rows can be set between 1 and 10. Drag Cols can be set between 1 and 15. The total number of slots in the Drag box will be, consequently, Drag Rows * Drag Cols. This property has a similar one for the Drop box 'Drop Rows – Drop Cols'.

Drag Empty Slot



This option sets texture to display in the Drag empty slots background. This property has a similar one for the Drop box 'Drop Empty Slot'.

Drag Filled Slot



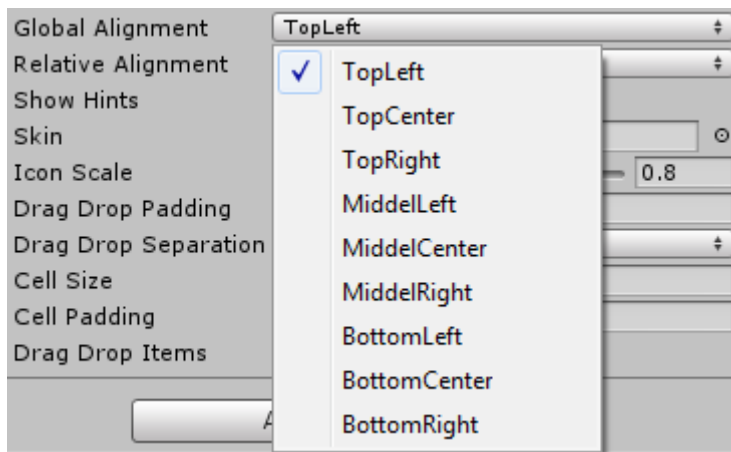
This option sets the texture to display in the Drag filled slots background. This property has a similar one for the Drop box 'Drop Filled Slot'.

Boxed Drag



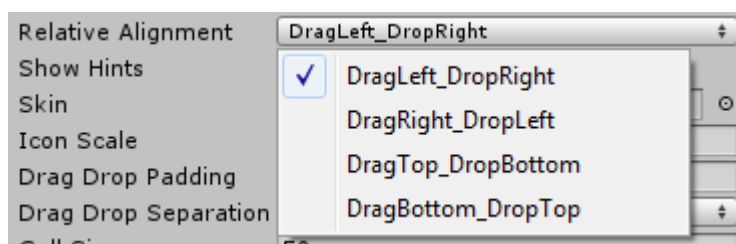
Set true to display a surrounding box for the drag box. This property has a similar one for the Drop box 'Boxed Drop'.

Global Alignment



This option sets the global alignment of the Drag and Drop System in the screen.

Relative Alignment



This option sets the alignment of the Drag and Drop relative to each other.

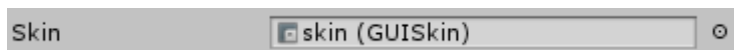
Show Hints



If set to on, when the mouse will be over an item, a ToolTip window will be displayed with the item info. See example:



Skin



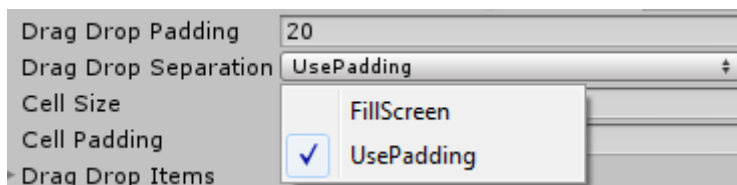
This option sets the Skin to be used in the ToolTip window. The asset will provide a default Skin but you can use any other or configure the existing one to match your look and feel.

Icon Scale



This option sets the Icon scale relative to cell. If your icons has no padding you can downsize it, or if your icons has a big padding you can oversize it.

Drag Drop Padding – Drag Drop Separation



Set the separation between the drag and drop boxes. If you set Separation to 'UsePadding' the space between Drag and Drop will be the Padding. If you set Separation to 'FillScreen' space between Drag and Drop will increase till both boxes where in the left and right of the screen (if relative alignment is set to left-right) or in the top and bottom of the screen (if relative alignment is set to top-down).

Cell Size



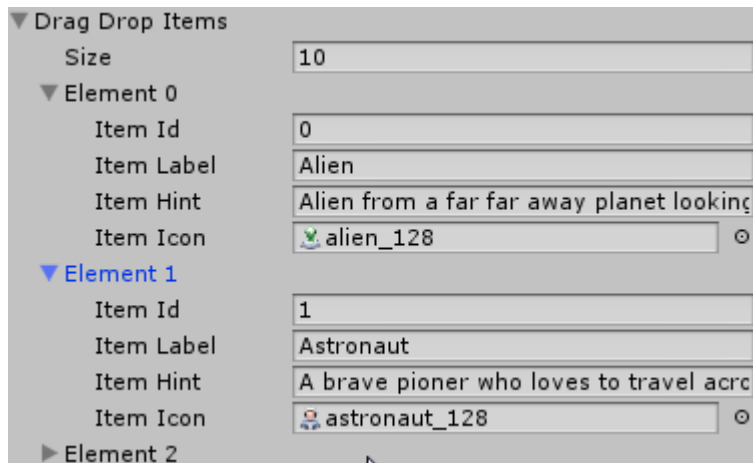
This option sets the size (width and height) for each slot in the Drag and Drop cells.

Cell Padding



This option sets the padding for each slot in the Drag and Drop cells.

Drag Drop Items



It is an array with all the items for the Drag Drop System. Each element needs to contain:

Item Id: Unique item identifier.

Item Label: The name f the item. This text will be the caption in the ToolTip window.

Item Hint: Description of the item. This text will be the ToolTip window.

Item Icon: This option sets the texture for this item.

Trigger Section

When the user try a drop the scrip will send a message to all the scripst in the object which has the method DropTrigger:

void DropTrigger (DragDropSystem dragdrop)

Also the script populate some public variables:

int originSlot: The slot index from where the item starts to be dragged

int destinationSlot: The slot where the item has been dropped

int droppedItemIndex: The index, in the items array, of the dropped item

bool originIsDrag: True if item comes from the drag box

bool destinationIsDrag: True if item is dropped in the drag box

DropErrorCode dropErrorCode: Error code of the resulting drop. The codes are

- noError: Drop successfully compelted
- notAllowedArea: Try to drop out of drag or drop areas
- notAllowedBox,: Try to drop in the drag area but no Bidirectional direction set
- notAllowedSlot: Try to drop in a filled slot but no MakeRoomfPossible behavior set

- noRoomForDrop: Try to drop in a completely filled drop area so cannot make any room

For example, you can create a script, attach it to the same object where the DragDropSystem is and create this method on it:

```
void DropTrigger (DragDropSystem dragdrop) {  
    Debug.Log ("DROP DETECTED");  
    Debug.Log ("Origin slot=" + dragdrop.originSlot + " Destination slot=" +  
    dragdrop.destinationSlot + " Item Index=" + dragdrop.droppedItemIndex);  
  
    if(dragdrop.originIsDrag)  
        Debug.Log("Origin=DRAG");  
    else  
        Debug.Log("Origin=DROP");  
  
    if(dragdrop.destinationIsDrag)  
        Debug.Log("Destination=DRAG");  
    else  
        Debug.Log("Destination=DROP");  
  
    Debug.Log ("ERROR CODE=" + dragdrop.dropErrorCode);  
}
```