

NGUI Action Bars

Contact Info: NGUIActionBars@gmail.com

Installation

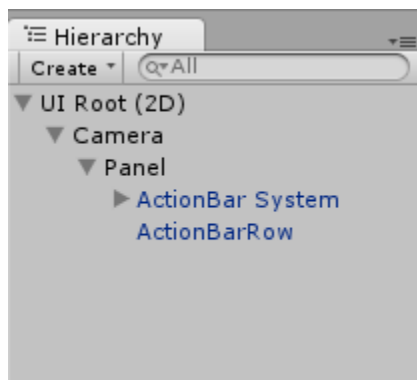
Import NGUI into your Unity Project. (Supported versions are 2.7 and 3.0)

Import Action Bars into your project

Create an (Action Bars/Resources/Prefabs) “ActionBar System” onto a UIPanel

Create an “ActionBarRow” onto a UIPanel (Does not have to be the same UIPanel as ActionBar System.

NOTE: Default Layer for this Plugin is Layer 8. Make sure your camera has the correct culling mask and event receiver mask set.



It should look similar to this

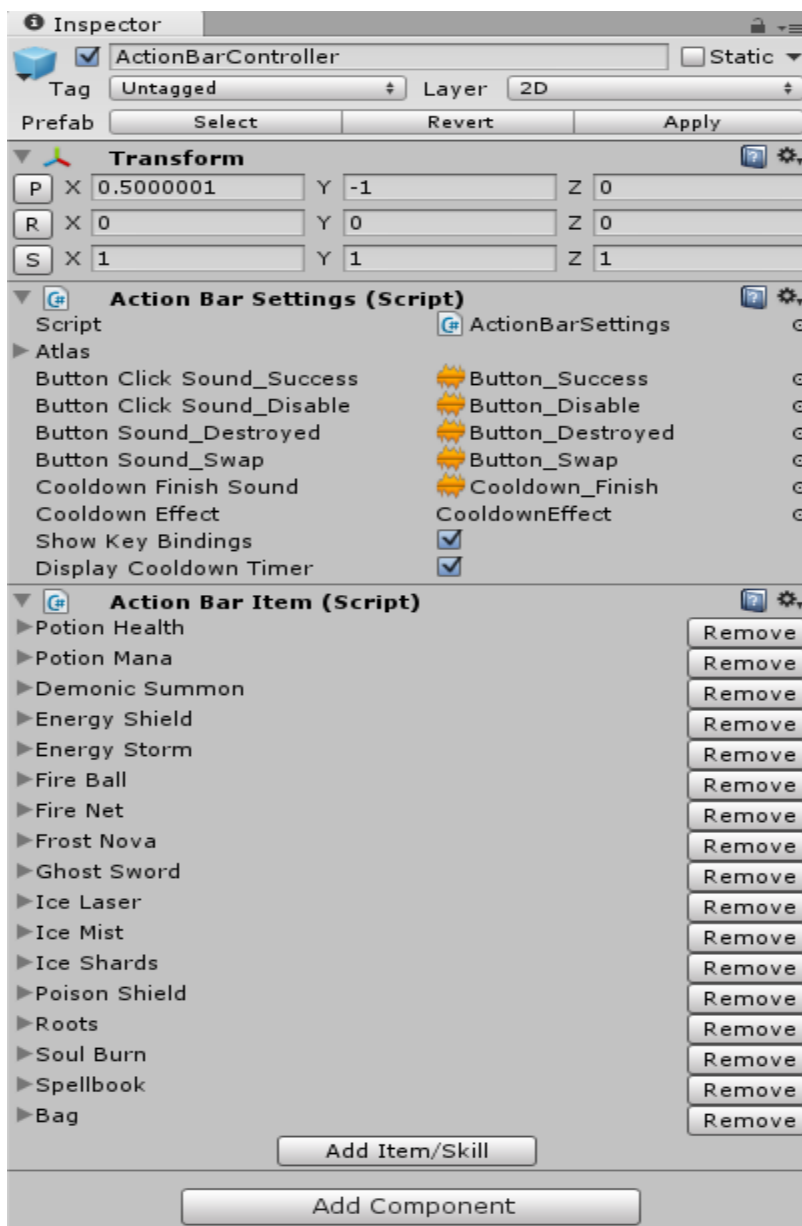
Settings

ActionbarController

ActionBarController inside of ActionBar System contains the textures that will used used by your buttons, the sounds and the Cooldown Effect.

-The textures are stored in an Atlas List.

-The Cooldown Effect is stored inside ActionBarSystem/Resources/Prefabs. Change the effects of the cooldown to whatever you see fit.



ActionBarController also stores the list of all your spells/items. Add a spell here by simply clicking the Add Item/Skill tab at the bottom. Upon adding an item it will become available to be set on a Button.

List of Settings for each Item:

Atlas: The atlas number set inside Action Bar Settings that this item will be using.

Icon: The name of the sprite inside the atlas that your item is using.

Group ID: Which Rows your item can be placed on. Ex: An item with ID 2 can be placed on a Row with ID 2, but not a Row with ID 0. (Rows can have multiple group IDs)

Cooldown: Cooldown duration of your Item.

Stackable: Whether your item has a limited amount of stacks.

Stack: How many starting stacks this item will have.

Display One Stack: If the item should show the stack counter at 1.

Destroy on Zero Stacks: If item is auto-destroyed upon reaching zero stacks.

Active Ability: Whether the item can be activated. (Passive skills should have this off)

Cooldown Animation: Whether to show the cooldown effect upon the cooldown finishing.

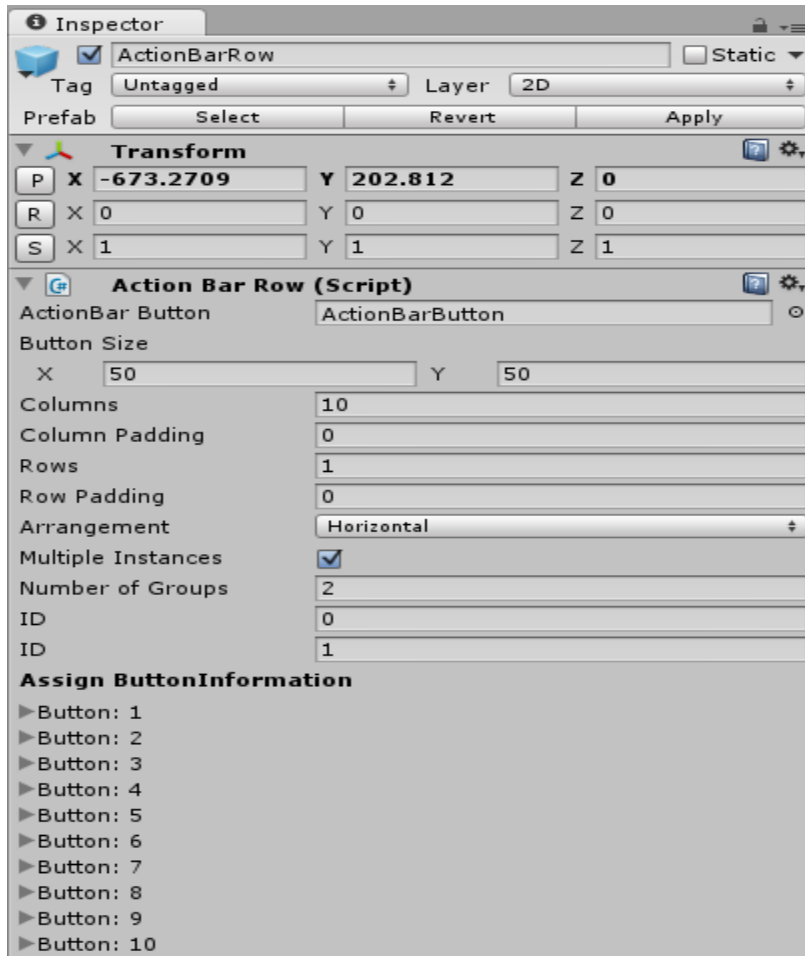
Cooldown Direction: Which direction the cooldown goes

Target: The GameObject that has a script function that this will call upon being clicked.

Function: The function name that is called on the Target GameObject.

ActionBarRow

This is what you use to create a row of buttons for your items/spells.



List of Settings

ActionBarButton: The button prefab that will be used as a base for the row of buttons. You can assign a different button for each row if you want a different looks for each button row. The base button prefab is stored in ActionbarSystem/Resources/Prefabs/Storage.

Button Size: The size of the buttons in the row.

Columns: How many columns should the row have?

Rows: How many rows should row have?

Column/Row padding: The padding in between each button for the Row/Column.

Arrangement: Which direction the buttons will be arranged.

Multiple Instances: If multiple of the same items/spells are allowed on this row.

Number of Groups: The amount of Group IDs this row will have.

ID: Which item group IDs to allow onto this row.

Action Bar Button

Individual settings for each button in the row.

Assign Button Information

▼ Button: 1

Hotkey: None

Locked: ☒

Clone on Pickup: ☐

Empty Button: ☐

Button Content

Potion Health

Seperate Instance: ☐

HotKey: The hotkey that will activate the button upon pressing. You can edit the Hotkey definitions/strings inside of the “ActionBarSettings” class if you want different values.

Locked: Whether the item can be picked up from the button.

Clone on Pickup: When picked up it will create a copy of the item inside the button.

Empty Button: If the button will start empty.

Button Content: The item or spell that will start in the button. These are created from the ActionBarController object.

Seperate Instance: Allows for multiple of the item/spell that share the same cooldown but *DO NOT* have the same stacks.