



# **Directed Navigation**

**Version: 1.4.0**

**Document version: 1.0**

**IEVO**

<b>OVERVIEW.....</b>	<b>5</b>
<b>SPECIFICATIONS.....</b>	<b>5</b>
<b>GET STARTED.....</b>	<b>6</b>
<b>MODES.....</b>	<b>9</b>
Automatic.....	9
Sector.....	11
Rectangle.....	12
RectTransform.....	13
SelectableList.....	15
Disabled.....	17
<b>ADVANCED SETTINGS.....</b>	<b>18</b>
Anchor.....	18
Identity.....	18
Shift.....	18
RectTransform.....	19
Omnidirectional.....	21
<b>DIRECTED NAVIGATION GRID.....</b>	<b>22</b>
Wrap Around.....	22
Disabled.....	22
Direct.....	23
Prev.....	25
Next.....	27
<b>API.....</b>	<b>28</b>
DirectedNavigation.....	28
Properties.....	28
• Active [Type: bool, Default value: true].....	28
• Selectable [Type: Selectable].....	28
• ConfigLeft [Type: Config].....	28
• ConfigRight [Type: Config].....	28
• ConfigUp [Type: Config].....	28
• ConfigDown [Type: Config].....	28
Events.....	28
• OnActive [Type: bool].....	28
Config.....	29
Properties.....	29
• Type [Type: DirectedNavigationType.Value, Default value: DirectedNavigationType.ValueAutomatic].....	29
• WrapAround [Type: bool].....	29
• UseEdges [Type: bool].....	29

• Omnidirectional [Type: bool].....	29
• Directivity [Type: float, Default: 0, Range: (-12) - 12].....	29
• Sector [Type: Config.SectorConfig].....	29
• Rectangle [Type: Config.RectangleConfig].....	29
• RectTransform [Type: Config.RectTransformConfig].....	29
• SelectableList [Type: Config.SelectableListConfig].....	29
• Anchor [Type: Config.AnchorConfig].....	29
Config.SectorConfig.....	30
Properties.....	30
• MinAngle [Type: float, Default: -25f, Range: MinAngleLimit - MaxAngle].....	30
• MinAngleLimit [Type: float, Readonly, Value: -90f].....	30
• MaxAngle [Type: float, Default: 25f, Range: MinAngle - MaxAngleLimit].....	30
• MaxAngleLimit [Type: float, Readonly, Value: 90f].....	30
• Radius [Type: float, Default: 500f].....	30
Config.RectangleConfig.....	30
Properties.....	30
• Verts [Type: Vector3[], Length: 4].....	30
Config.RectTransformConfig.....	30
Properties.....	31
• RectTransform [Type: RectTransform, Default: null].....	31
Config.AnchorConfig.....	31
Properties.....	31
• Type [Type: Anchor.Type, Values: Identity, Shift, RectTransform].....	31
• Shift [Type: Vector3].....	31
• RectTransform [Type: RectTransform].....	31
Methods.....	31
• GetAnchoredPositionLocal (Vector3 positionOfObject, Transform space).....	31
• SetShift (Vector3 value).....	31
• SetRectTransform (RectTransform value).....	31
• Reset().....	31
Config.SelectableListConfig.....	32
Properties.....	32
• SelectableList [Type: Selectable[]].....	32
DirectedNavigationType.Value [Enum:int].....	32
• Disabled.....	32
• Automatic.....	32
• Sector.....	32
• Rectangle.....	32
• RectTransform.....	32

• SelectableList.....	32
DirectedNavigationGrid.....	33
Properties.....	33
• GridLayoutGroup [Type: GridLayoutGroup].....	33
• DirectedNavigations [Type: List<DirectedNavigation>].....	33
• ConfigLeft [Type: Config].....	33
• ConfigRight [Type: Config].....	33
• ConfigUp [Type: Config].....	33
• ConfigDown [Type: Config].....	33
Events.....	33
• OnUpdateGrid.....	33
Methods.....	33
• UpdateGrid().....	33
GridSideConfig.....	34
Properties.....	34
• OutgoingTransitions [Type: bool].....	34
• WrapAround [Type: ].....	34
• Strict [Type: bool].....	34
• Loop [Type: bool].....	34
• Directivity [Type: float, Default: 0, Range: (-12) - 12].....	34
WrapAroundType [Enum:int].....	34
• Disabled.....	34
• Direct.....	34
• Prev.....	35
• Next.....	35
<b>TIPS AND TRICKS.....</b>	<b>36</b>
Grid of elements with transitions only between items.....	36
Option 1 (Preferable).....	37
Option 2.....	39
Grid with Wrap Around.....	41
Simple grid with Wrap Around.....	41
Grid with Wrap Around and transition by lines.....	43
Explicit.....	45
<b>CONTACTS.....</b>	<b>47</b>

## OVERVIEW

**uGUI Directed Navigation** for Unity is a tool which provides extended UI navigation options to control how keyboard/gamepad navigate through UI elements. It has improved automatic mode, multiple extended modes and also provides the ability to control navigation in four directions separately (Left, Right, Up, Down).

From usage point of view it is a component, which is added to any GameObject which has component inherited from Selectable - Button, Toggle, Slider etc.

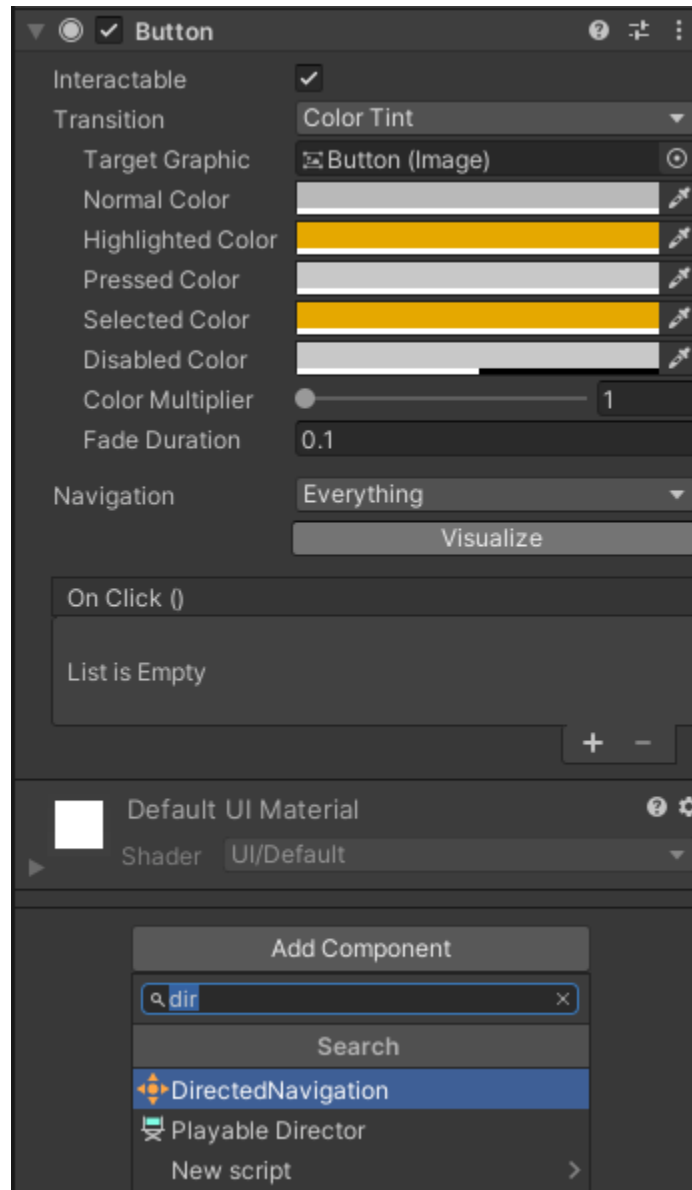
## SPECIFICATIONS

Pure C#. As a result, support any platform.

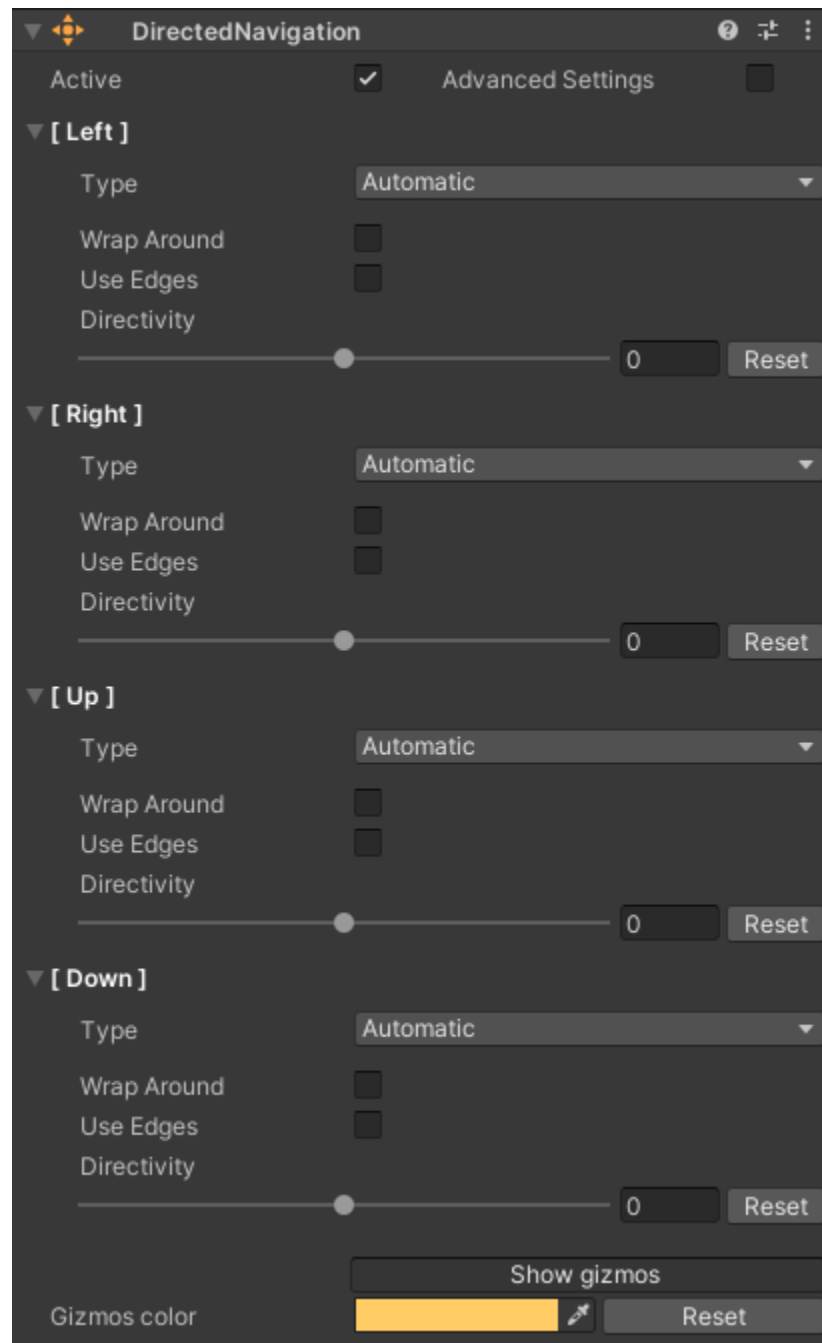
Compatible with Unity 2019.4 and above.

## GET STARTED

To enable **uGUI Directed Navigation** for the Selectable (or a class inherited from it) add a component **DirectedNavigation** to GameObject.



Just added component look like listed below.



Property **Active** can be used for activation/deactivation of the **DirectedNavigation**. This property also can be changed in runtime through API.

The **Advanced Settings** option activates additional options (**Anchor**, **Omnidirectional**) that can be useful in non-standard situations.

Each direction (**Left**, **Right**, **Up**, **Down**) of navigation can be controlled separately, it can be switched to different modes or disabled.

Button **Show gizmos** enable/disable gizmos. You can choose the color of gizmos in color picker **Gizmos color**, according to your project needs.

**Important:** **uGUI Directed Navigation** is a replacement for the **Navigation** section of **Selectable** and all navigation settings must be done through this component. All settings in the **Navigation** section of **Selectable** will be ignored.

**Important:** Gizmos can be disabled if in the **Selectable** (Button etc.) component is switched off the **Visualize** button, this option has higher priority than button **Show gizmos**.

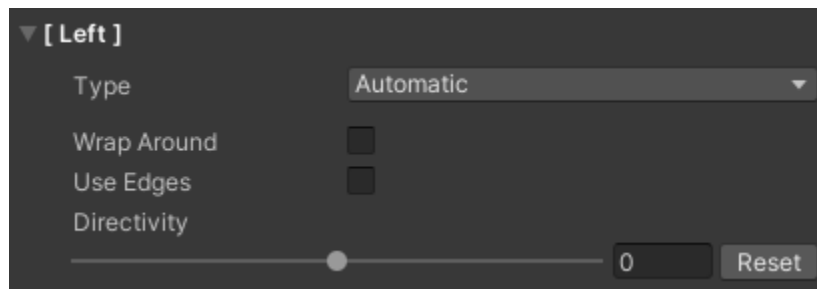
**Important:** When the **DirectedNavigation** component is added you can notice that the **Navigation** section of **Selectable** is switched to **Explicit** mode. This is required for drawing navigation visualization lines and used only in Editor. In Runtime this will be just ignored and all navigation handling will be done by the **DirectedNavigation** component. This implementation is due to limitations of the Unity editor. If you need **Explicit** mode you can use **DirectedNavigation** with **SelectableList** mode with one element in the list.



## MODES

### Automatic

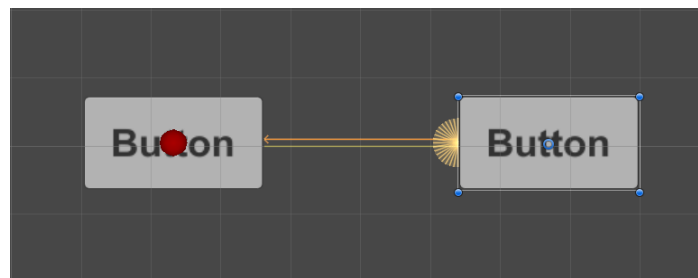
This navigation mode is similar to the Automatic of Selectable component, but has an improved algorithm of work, and also provides options to adjust it.



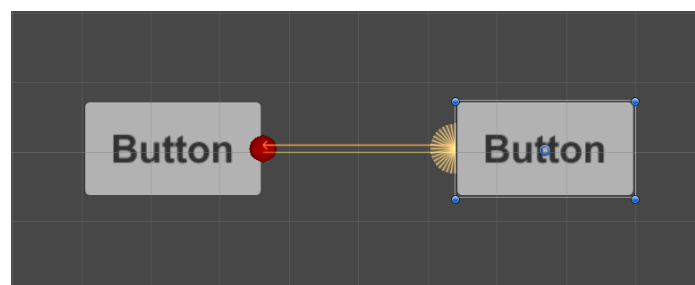
**Wrap Around** allows for navigation to wrap around to the farthest element in the opposite direction of movement from the current element.

**Use Edges** option is define which part of searched objects will be used, center or edges. Schematically it looks like listed below.

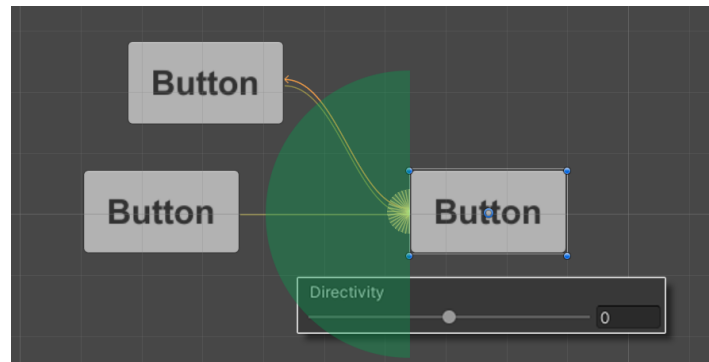
**Use Edges** disabled:



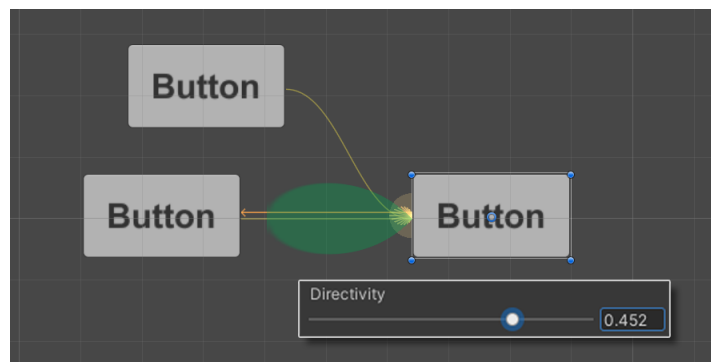
**Use Edges** enabled:



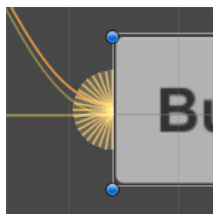
**Directivity** property. By default, the search for elements in the current direction is carried out in a sector of 180 degrees. All objects have the same priority and the closest object will be selected for the navigation. With the **Directivity** property we can control the priority of getting objects depending from the angle to the current direction. We can visualize it like listed below. Green sector displays priority.



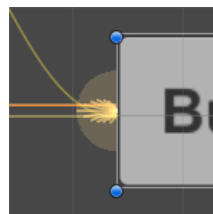
If we increase the **Directivity**, this will lower the priority of objects that are located with a deviation from the current direction. We can visualize it like this.



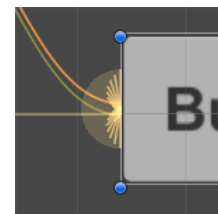
The current state of **Directivity** shows gizmos.



Directivity: 0



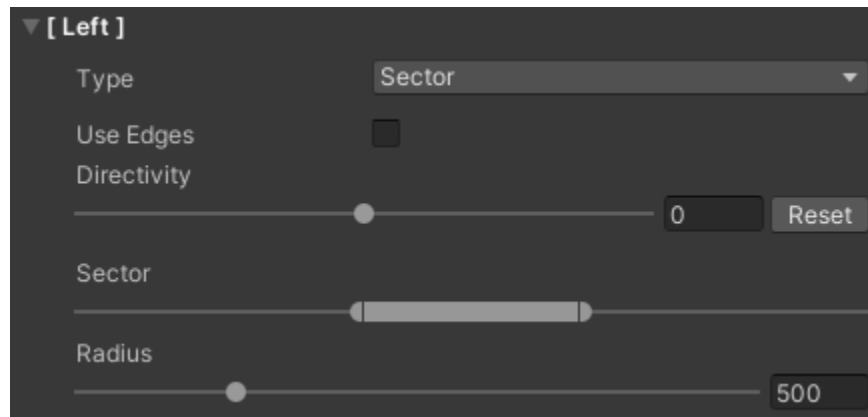
Directivity: 0.5



Directivity: -0.5

## Sector

For situations where the selection of an element to transition should be constrained by a radius or an angle.



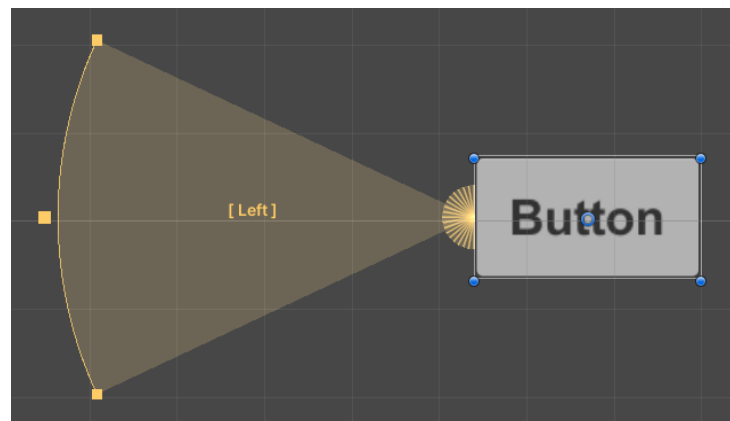
**Use Edges** option is define which part of searched objects will be used, center or edges. For more information look at this [topic](#).

**Directivity** controls the priority of getting objects depending from the angle to the current direction. For more information look at this [topic](#).

**Sector** option defines a sector, start angle, end angle, and direction.

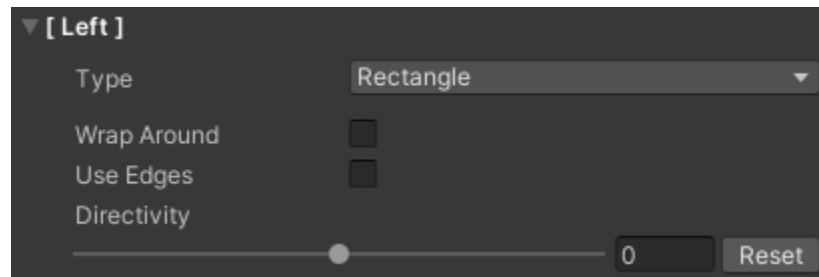
**Radius** option defines a radius of the sector.

Sector params can be also changed with gizmos. Just drag square elements on the edge of the sector.



## Rectangle

Restricts the element's search area to a rectangle. All UI elements outside the rectangle will be ignored.

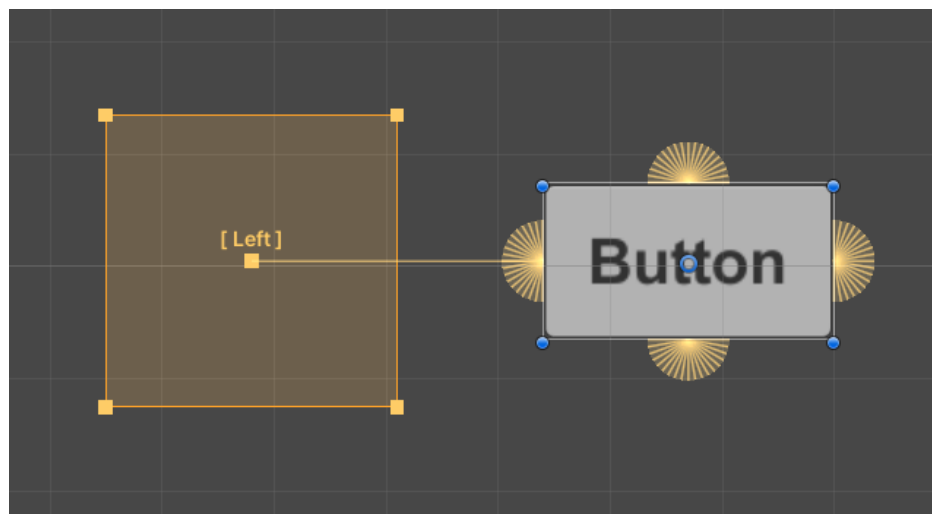


**Wrap Around** allows for navigation to wrap around to the farthest element in the opposite direction of movement from the current element.

**Use Edges** option is define which part of searched objects will be used, center or edges. For more information look at this [topic](#).

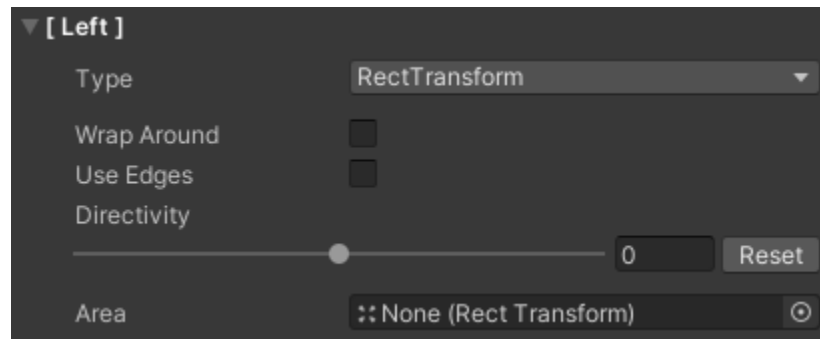
**Directivity** controls the priority of getting objects depending from the angle to the current direction. For more information look at this [topic](#).

To change the size and position of the rectangle use gizmos. Just drag the square elements on the edge of the rectangle or in the center.



## RectTransform

Restricts the element's search area to a bounds of the 'RectTransform' component. All UI elements outside the bounds will be ignored.



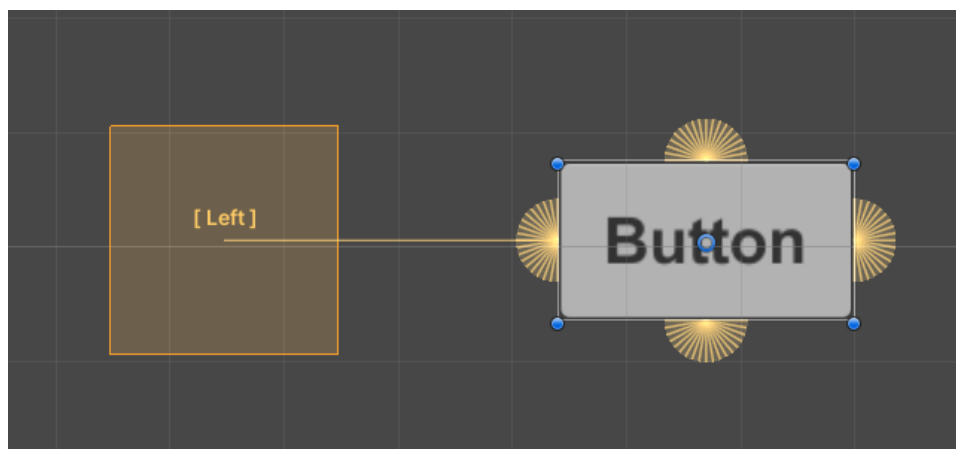
**Wrap Around** allows for navigation to wrap around to the farthest element in the opposite direction of movement from the current element.

**Use Edges** option is define which part of searched objects will be used, center or edges. For more information look at this [topic](#).

**Directivity** controls the priority of getting objects depending from the angle to the current direction. For more information look at this [topic](#).

**Area** is a RectTransform component which will be used.

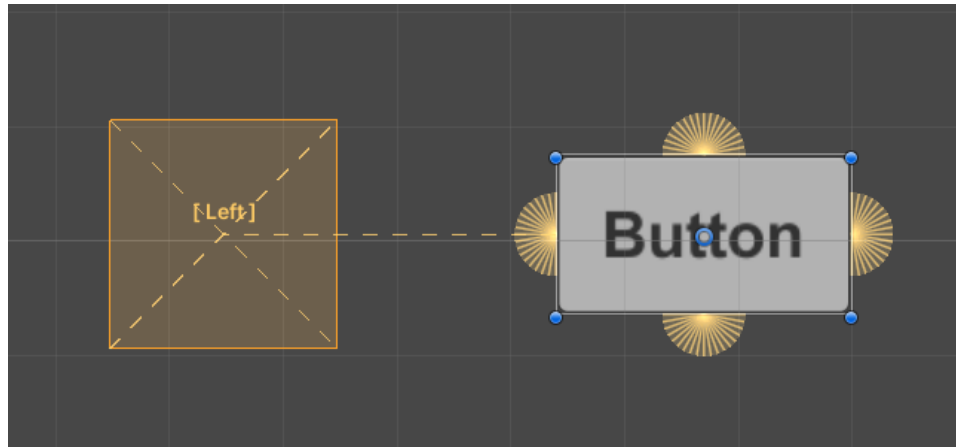
In the Scene it looks like this.



RectTransform can be assigned or changed at runtime.

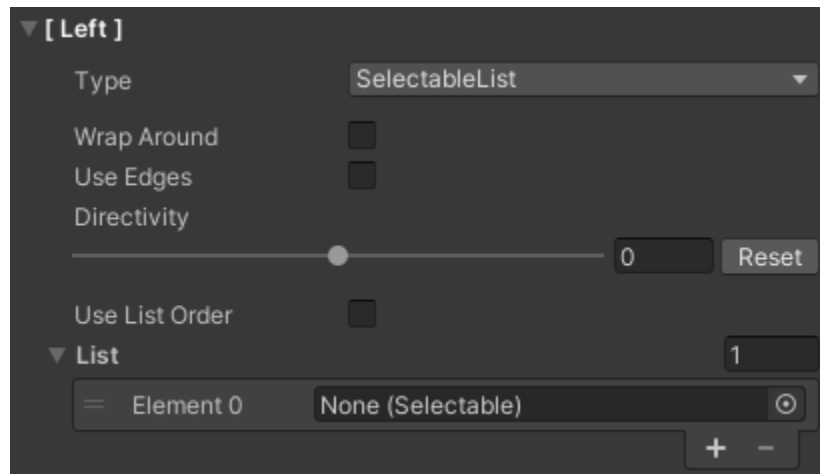
If RectTransform will be disabled behavior in this direction will be similar to **Disabled** mode. No navigation in this direction.

In Scene view disabled RectTransform will be marked with dotted lines.



## SelectableList

The selection of an element for the transition is carried out not from the entire scene, but from a list defined by the developer.



**Wrap Around** allows for navigation to wrap around to the farthest element in the opposite direction of movement from the current element.

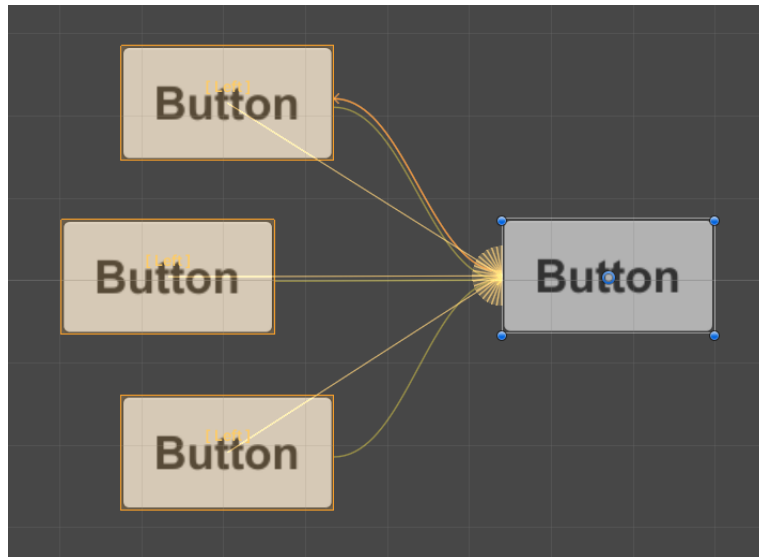
**Use Edges** option is define which part of searched objects will be used, center or edges. For more information look at this [topic](#).

**Directivity** controls the priority of getting objects depending from the angle to the current direction. For more information look at this [topic](#).

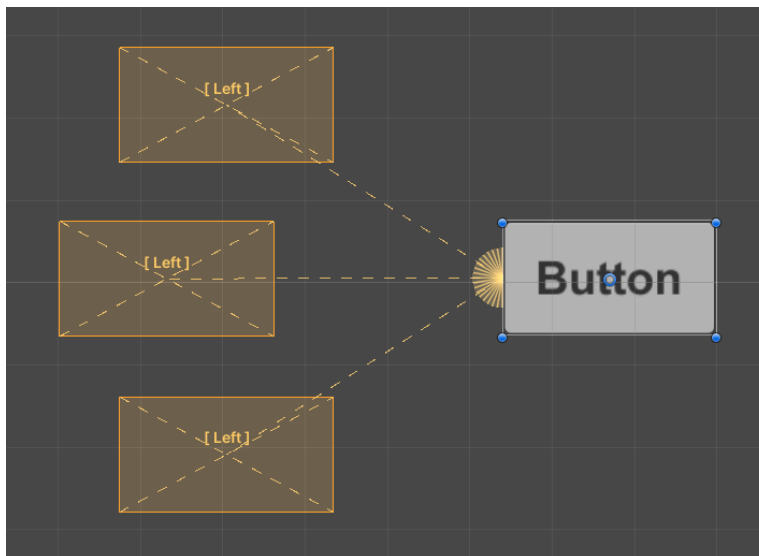
**Use List Order** Use List Order if enabled all navigation algorithms will be ignored and transition to elements will be according to their order in the list. If the first element for transition will be disabled in the hierarchy transition will be executed to the next element in the list. Note: In some cases required to enable **Omnidirectional** option (Advanced Settings).

**List** is a list of Selectables.

In the Scene it looks like this.



If some of the objects from the list will be disabled these objects will be ignored during selection of objects to navigate. Also in Scene view these objects will be marked with dotted lines.





## Disabled

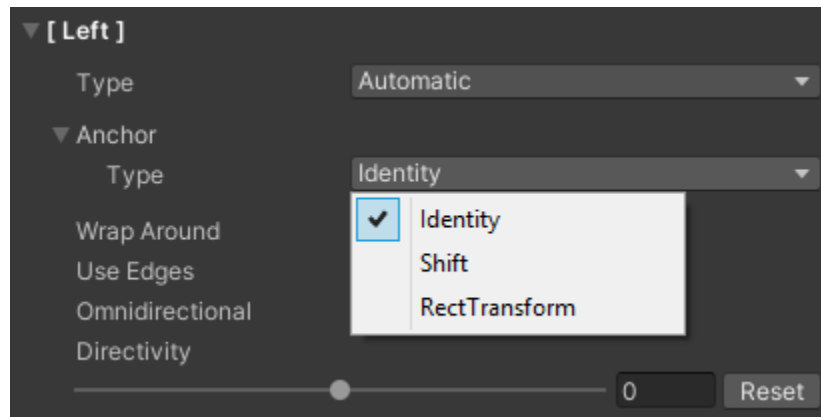
No navigation in this direction.

## ADVANCED SETTINGS

The **Advanced Settings** option includes additional options, such as **Anchor** and **Omnidirectional**, which can be useful in non-standard situations.

### Anchor

Allows you to shift the point from which the search for an element to transition occurs.

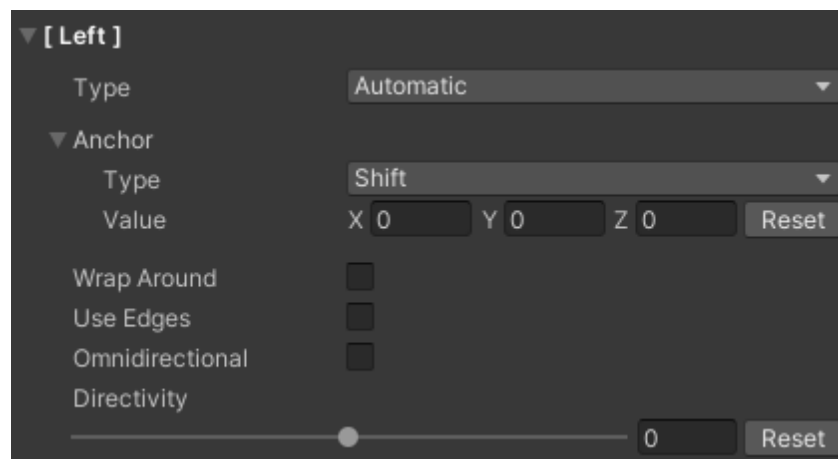


#### Identity

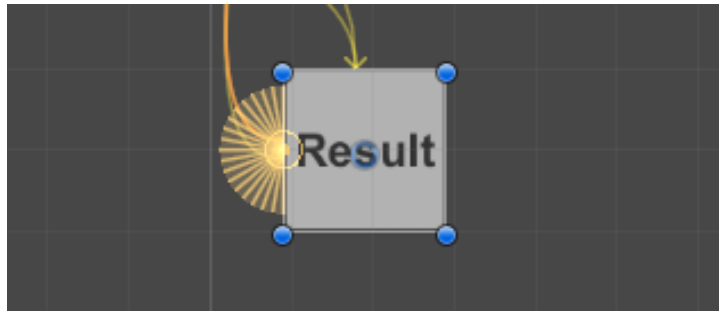
Anchor position unchanged.

#### Shift

Allows to move the **Anchor** arbitrarily relative to the current object.



When selecting this option, a circle will appear around the **Directivity** gizmo, which can be used to move the anchor by mouse.

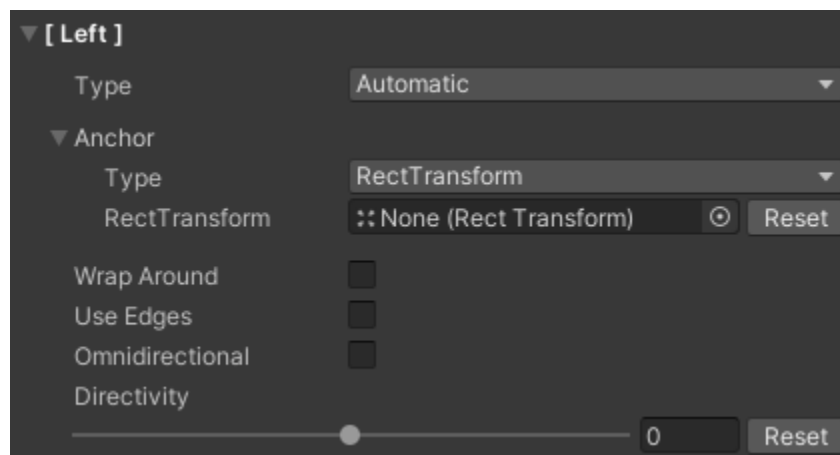


Also the position of Shift can be set through the **API**.

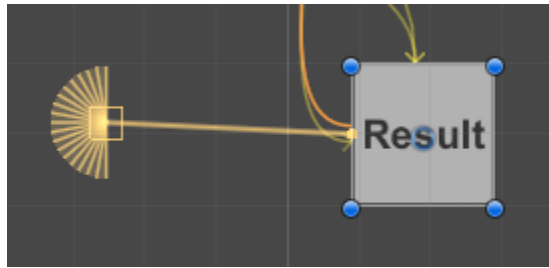
## RectTransform

Allows you to use a **RectTransform** as a position point from which the search for an element to transition occurs.

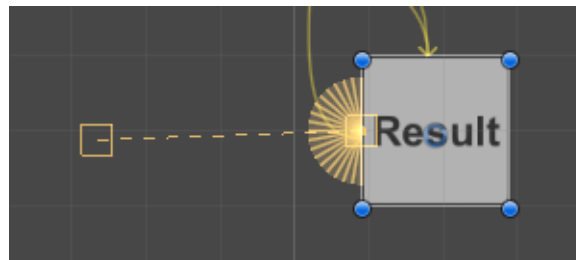
If **RectTransform** becomes inactive a position point will be the same as Identity.



When selecting this option, a square will appear around the **Directivity** gizmo.



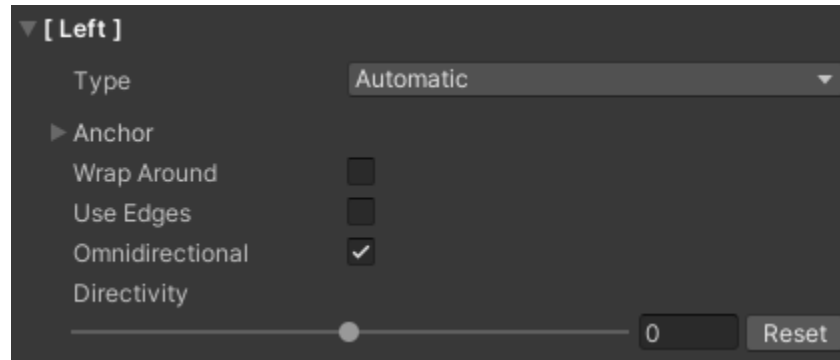
If the source **RectTransform** will be deactivated, gizmos will show it like in the image below.



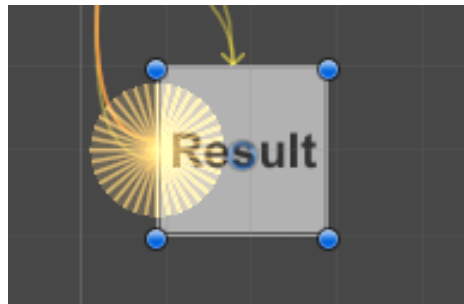
Also the **RectTransform** can be set through the **API**([RectTransform](#) ).

## Omnidirectional

By default, the navigation for each side only occurs in the direction it is facing, which means that elements on the other side are not used for transitions under any circumstances. **Omnidirectional** option allows you to bypass this limitation.

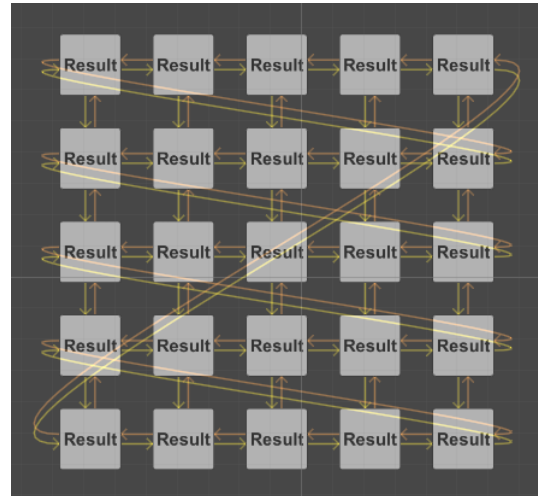
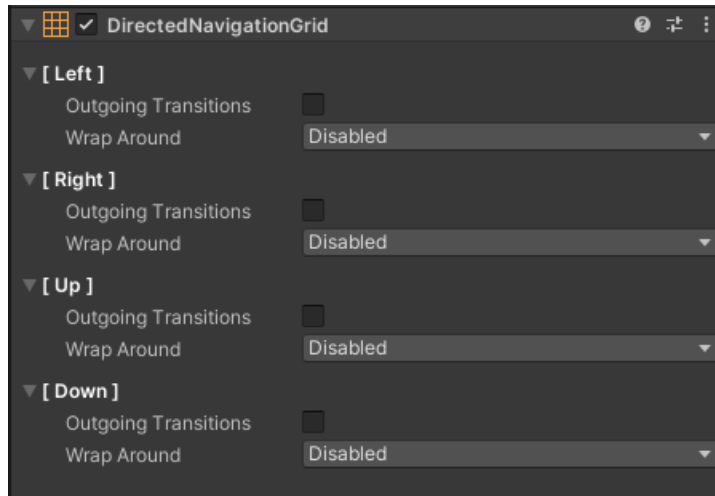


When enabled, the gizmo on the stage will also change, it will begin to display omnidirectionally.



## DIRECTED NAVIGATION GRID

**Directed Navigation Grid** additional component which comes with **uGUI Directed Navigation**. This component is required for fast and convenient construction of grid structures with predictable transitions between elements. For its operation, the **GridLayoutGroup** component is required, as well as the presence of the **DirectedNavigation** component on child objects.



You can set individual settings for each side of the grid.

**Outgoing Transition** allows/disallows jumping to objects that are not part of the grid.

### Wrap Around

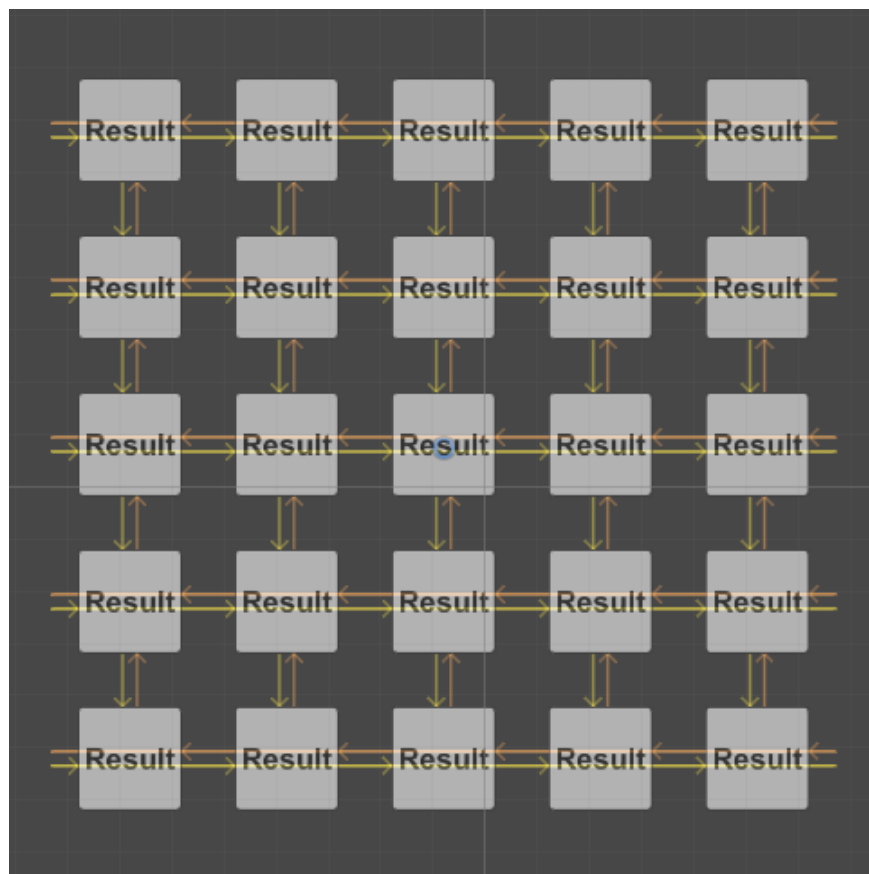
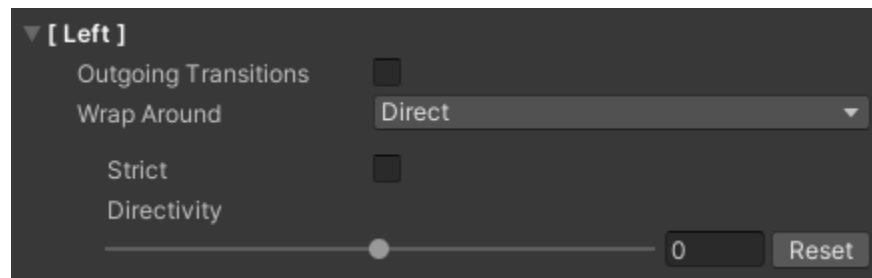
Determines how the transition from the extreme elements to the opposite side of the grid will be carried out, provides several modes of operation (**Disabled**, **Direct**, **Prev**, **Next**).

#### Disabled

No **Wrap Around** transition for this side.

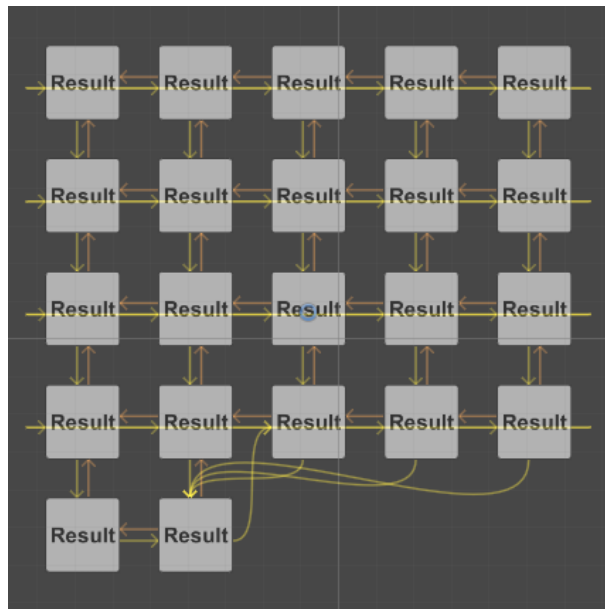
## Direct

The transitions are made in a straight line.

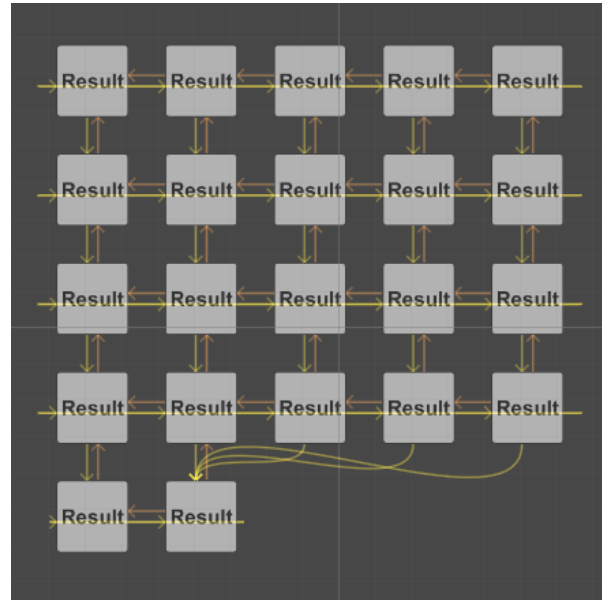


**Strict** option is needed in incomplete grids, when one of the rows or columns of elements is less than the grid dimension. To better understand this option, let's look at an example.

First let's look at the grid with disabled **Strict** option.



We can see that the transition from the last element in the last row goes to upper elements. Now let's enable the **Strict** option.



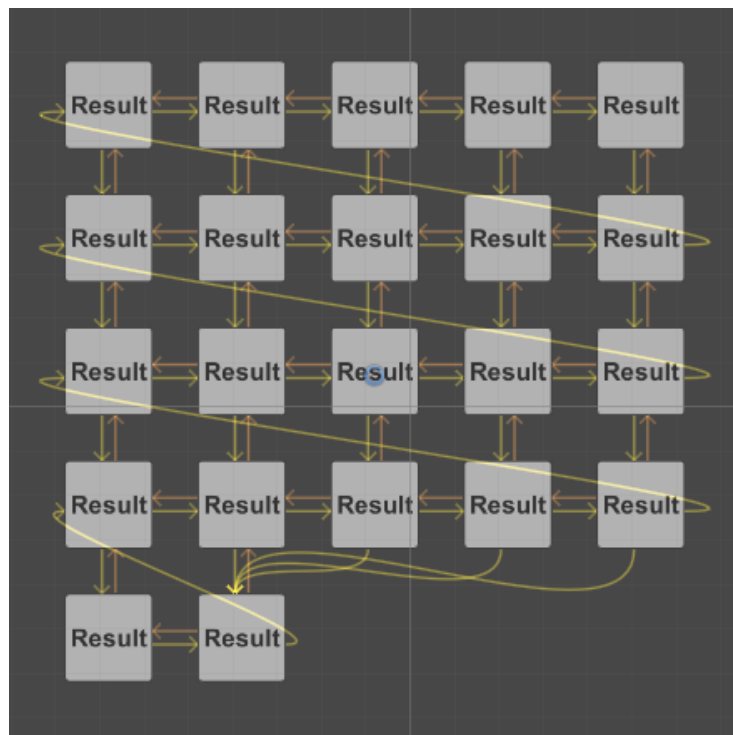
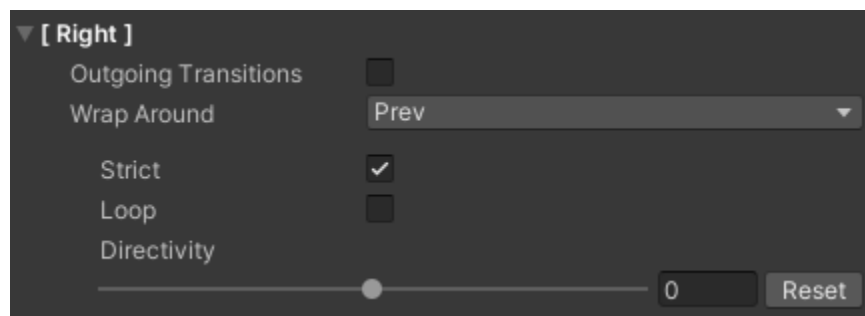
Transition from the last element in the last row goes in the opposite direction to the first element.



The **Directivity** option has the same functionality as it in the **DirectedNavigation**. See [Directivity](#).

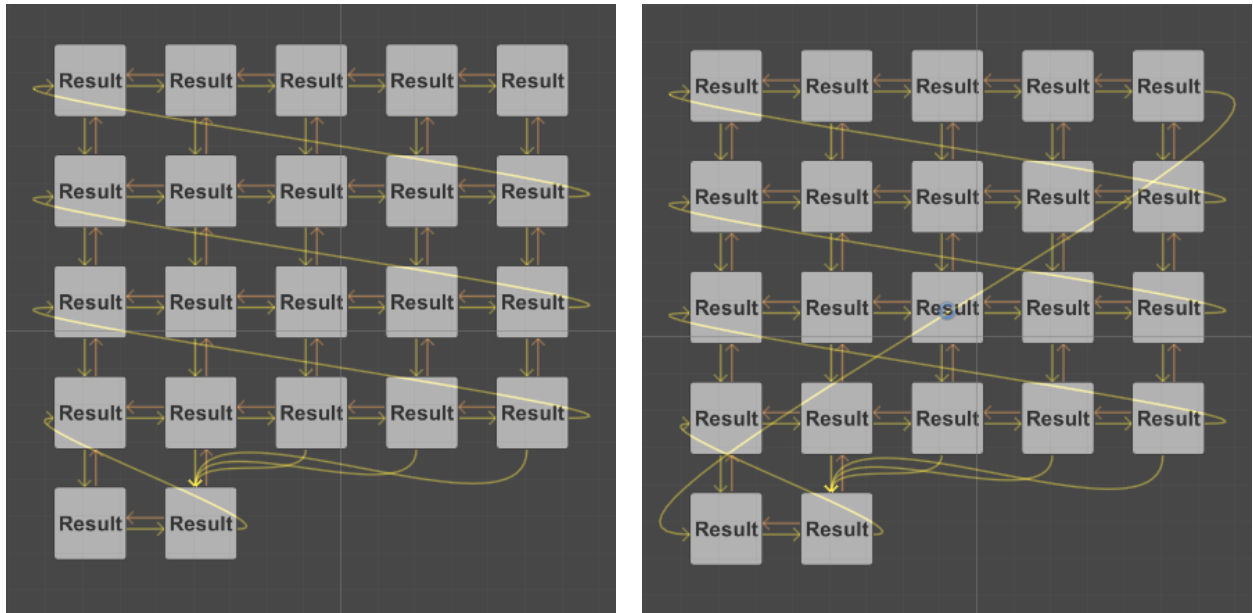
## Prev

The transition is carried out to the previous line/column.



**Strict** option is needed in incomplete grids, when one of the rows or columns of elements is less than the grid dimension. Prevents transition to near elements. See [more](#).

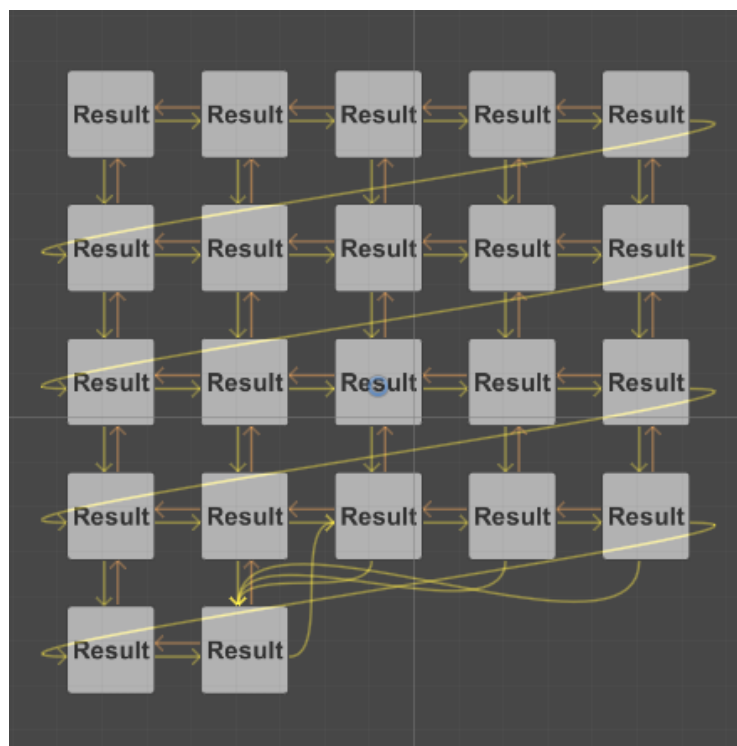
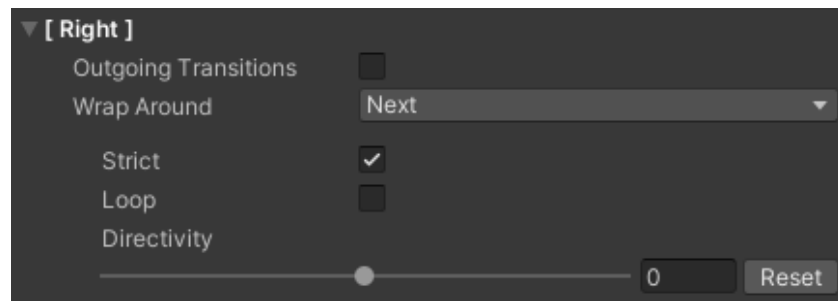
**Loop** enable/disable transition from last(first) element in the grid to the first(last) element.



The **Directivity** option has the same functionality as it in the **DirectedNavigation**. See [Directivity](#).

## Next

The transition is carried out to the next line/column.



**Strict** option is needed in incomplete grids, when one of the rows or columns of elements is less than the grid dimension. Prevents transition to near elements. See [more](#).

**Loop** enable/disable transition from last(first) element in the grid to the first(last) element. See [more](#).

The **Directivity** option has the same functionality as it in the **DirectedNavigation**. See [Directivity](#).

## API

### DirectedNavigation

Main component. Inherits interface [IMoveHandler](#).

#### Properties

- **Active** [Type: bool, Default value: true]  
Enable/Disable UI navigation handling.
- **Selectable** [Type: Selectable]  
Selectable to which DirectedNavigation added.
- **ConfigLeft** [Type: [Config](#)]  
Configuration for the Left side.
- **ConfigRight** [Type: [Config](#)]  
Configuration for the Right side.
- **ConfigUp** [Type: [Config](#)]  
Configuration for the Up side.
- **ConfigDown** [Type: [Config](#)]  
Configuration for the Down side.

#### Events

- **OnActive** [Type: bool]  
Notifies about on/off **Direction Navigation**.

## Config

Contains configuration for the side.

### Properties

- **Type** [Type: [DirectedNavigationType.Value](#), Default value: [DirectedNavigationType.ValueAutomatic](#)]  
Type of DirectedNavigation.
- **WrapAround** [Type: bool]  
Allows for navigation to wrap around to the farthest element in the opposite direction of movement from the current element.
- **UseEdges** [Type: bool]  
Define which part of searched objects will be used, center or edges.
- **Omnidirectional** [Type: bool]  
Allows for navigation elements from different directions. For more information see [Omnidirectional](#).
- **Directivity** [Type: float, Default: 0, Range: (-12) - 12]  
For more information see [Automatic](#).
- **Sector** [Type: [Config.SectorConfig](#)]  
Configuration for the Sector mode.
- **Rectangle** [Type: [Config.RectangleConfig](#)]  
Configuration for the Rectangle mode.
- **RectTransform** [Type: [Config.RectTransformConfig](#)]  
Configuration for the RectTransform mode.
- **SelectableList** [Type: [Config.SelectableListConfig](#)]  
Configuration for the SelectableList mode.
- **Anchor** [Type: [Config.AnchorConfig](#)]  
Configuration of Anchor. For more information see [Anchor](#).

## Config.SectorConfig

Contains configuration of [Sector](#) mode.

### Properties

- **MinAngle** [Type: float, Default: -25f, Range: MinAngleLimit - MaxAngle]  
Left angle of the sector.
- **MinAngleLimit** [Type: float, Readonly, Value: -90f]  
MinAngle limit.
- **MaxAngle** [Type: float, Default: 25f, Range: MinAngle - MaxAngleLimit]  
Right angle of the sector.
- **MaxAngleLimit** [Type: float, Readonly, Value: 90f]  
MaxAngle limit.
- **Radius** [Type: float, Default: 500f]  
Radius of the sector.

## Config.RectangleConfig

Contains configuration of [Rectangle](#) mode.

### Properties

- **Verts** [Type: Vector3[], Length: 4]  
Vertices of the rectangle. Coordinates in local space.

## Config.RectTransformConfig

Contains configuration of [RectTransform](#) mode.

## Properties

- **RectTransform** [Type: RectTransform, Default: null]

Contains a link on the RectTransform component, bounds of which restricts the element's search area.

## Config.AnchorConfig

Contains configuration of [Anchor](#).

## Properties

- **Type** [Type: Anchor.Type, Values: **Identity**, **Shift**, **RectTransform**]

Anchor type.

- **Shift** [Type: Vector3]

Anchor shift relatively to the current object, local coordinate space used.

- **RectTransform** [Type: RectTransform]

RectTransform position of which is used as Anchor.

## Methods

- **GetAnchoredPositionLocal** (Vector3 positionOfObject, Transform space)

Returns anchored position of object.

- **SetShift** (Vector3 value)

Sets Anchor shift.

- **SetRectTransform** (RectTransform value)

Sets as Anchor RectTransform.

- **Reset()**

Resets anchor.

## Config.SelectableListConfig

Contains configuration of [SelectableList](#) mode.

### Properties

- **SelectableList** [Type: Selectable[]]

List of Selectables from which will be selected element for navigation.

## DirectedNavigationType.Value [Enum:int]

Modes of DirectedNavigation.

- **Disabled**  
[Navigation disabled.](#)
- **Automatic**  
[Automatic navigation.](#)
- **Sector**  
[Sector mode.](#)
- **Rectangle**  
[Rectangle mode.](#)
- **RectTransform**  
[RectTransform mode.](#)
- **SelectableList**  
[SelectableList mode.](#)



## DirectedNavigationGrid

Component responsible for grids. Inherits interface [ILayoutController](#).

### Properties

- **GridLayoutGroup** [Type: [GridLayoutGroup](#)]  
The Grid Layout Group component places its child layout elements in a grid.
- **DirectedNavigations** [Type: List<[DirectedNavigation](#)>]  
List of [DirectedNavigation](#)s controlled by DirectedNavigationGrid.
- **ConfigLeft** [Type: [Config](#)]  
Configuration for the Left side.
- **ConfigRight** [Type: [Config](#)]  
Configuration for the Right side.
- **ConfigUp** [Type: [Config](#)]  
Configuration for the Up side.
- **ConfigDown** [Type: [Config](#)]  
Configuration for the Down side.

### Events

- **OnUpdateGrid**  
Notifies about grid update.

### Methods

- **UpdateGrid()**  
Rebuild grid structure of grid according GridLayoutGroup. In most cases this method is called by the internal mechanism of **DirectedNavigationGrid** and not required to call it manually.

## GridSideConfig

Contains configuration for side in [DirectedNavigationGrid](#).

### Properties

- **OutgoingTransitions** [Type: bool]

Outgoing Transition allows/disallows jumping to objects that are not part of the grid.

- **WrapAround** [Type: ]

Determines how the transition from the extreme elements to the opposite side of the grid will be carried out, provides several modes of operation (Disabled, Direct, Prev, Next). See [more](#).

- **Strict** [Type: bool]

Strict option is needed in incomplete grids, when one of the rows or columns of elements is less than the grid dimension.

- **Loop** [Type: bool]

Loop enable/disable transition from last(first) element in the grid to the first(last) element.

- **Directivity** [Type: float, Default: 0, Range: (-12) - 12]

For more information see [Automatic](#).

## WrapAroundType [Enum:int]

In **DirectedNavigationGrid** determines the mode of **Wrap Around**.

- **Disabled**

No Wrap Around transition for this side.

- **Direct**

The transitions are made in a straight line.

- **Prev**

The transition is carried out to the previous line/column.

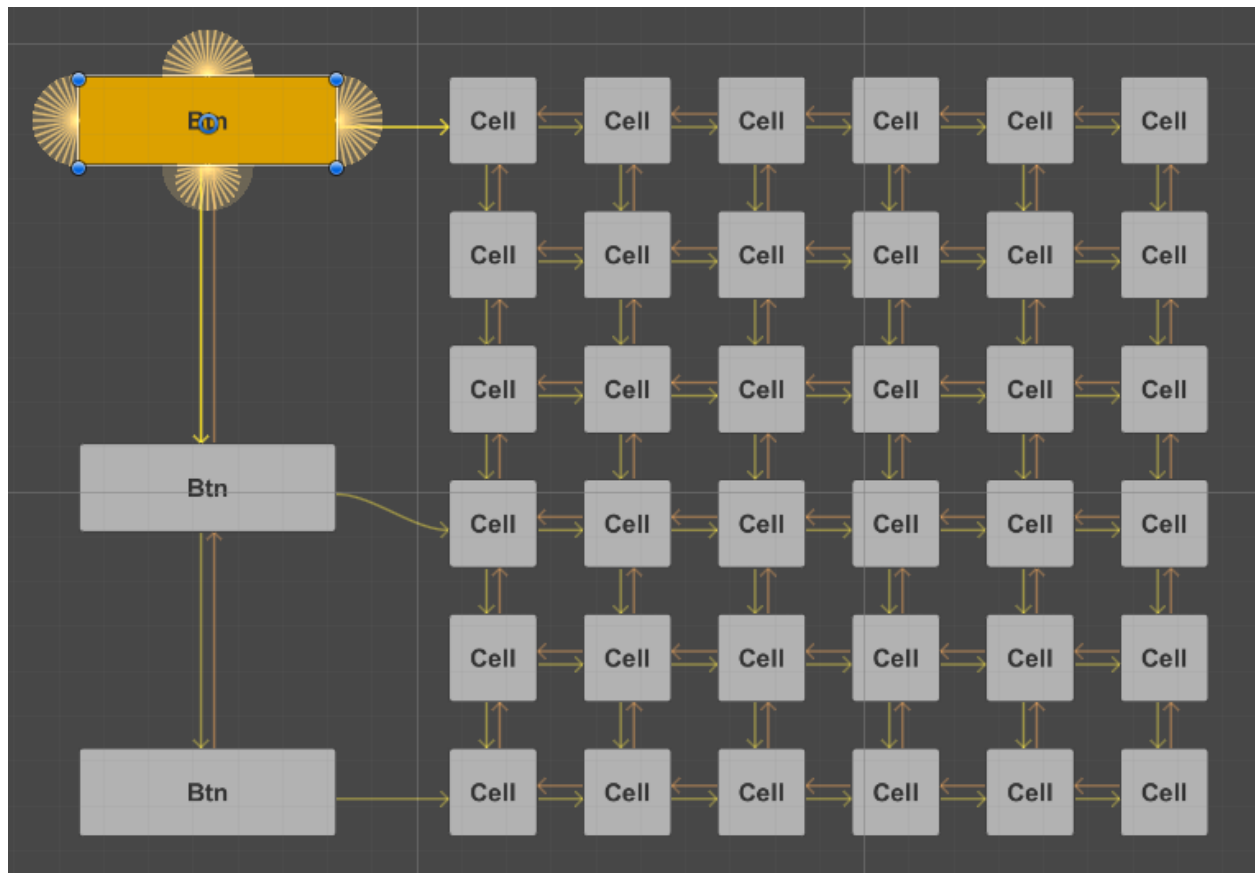
- **Next**

The transition is carried out to the next line/column.

## TIPS AND TRICKS

### Grid of elements with transitions only between items

Implement a grid of dynamically created elements (from prefab) with transitions only between items of the grid possible in different ways. Below are a couple of options, both options require little coding. For each of these options you can **download** ready examples.



**Important:** Note that for grids recommended usage of [DirectedNavigationGrid](#).

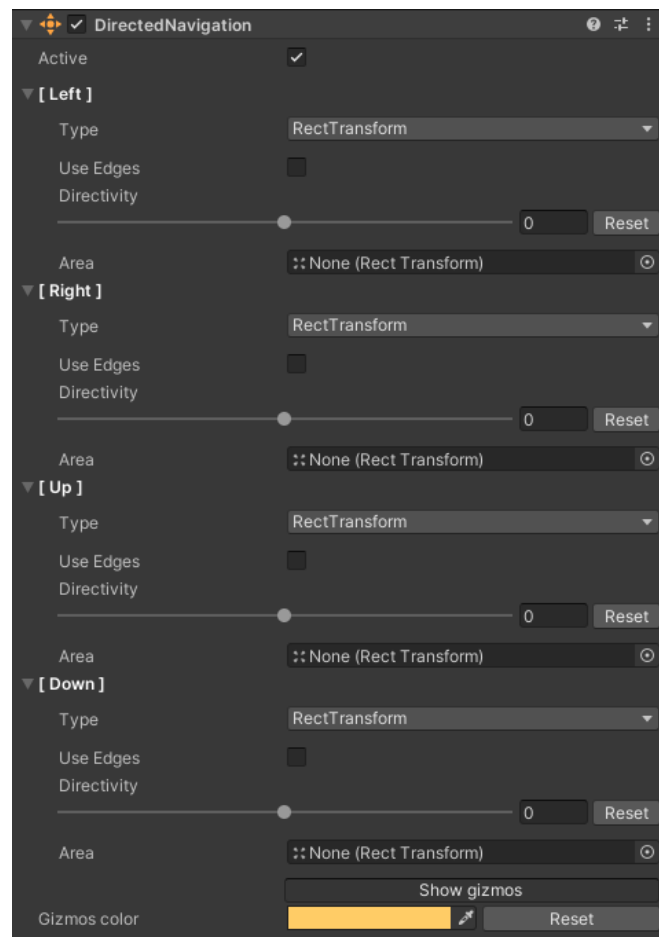
## Option 1 (Preferable)

### Download link:

[https://ievo.games/uGUIDirectedNavigation/Downloads/Examples/LimitedGrid\\_1.unitypackage](https://ievo.games/uGUIDirectedNavigation/Downloads/Examples/LimitedGrid_1.unitypackage)

This option uses a [RectTransform](#) mode.

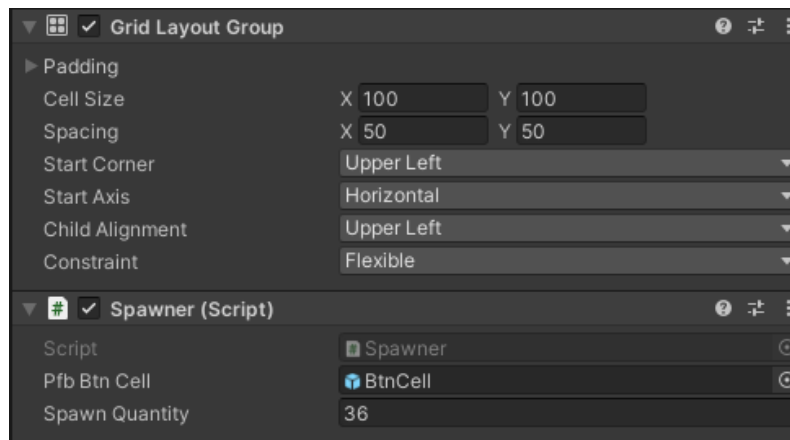
Create a grid item prefab and add **DirectedNavigation** component. For all sides, set the **RectTransform** type.



**Note:** Navigation mode also possible change from code:

```
dirNav.ConfigDown.Type = DirectedNavigationType.Value.RectTransform;  
dirNav.ConfigUp.Type = DirectedNavigationType.Value.RectTransform;  
dirNav.ConfigLeft.Type = DirectedNavigationType.Value.RectTransform;  
dirNav.ConfigRight.Type = DirectedNavigationType.Value.RectTransform;
```

Create a GameObject container for grid items and add to it **GridLayoutGroup** component.



Create and add to the grid items container component **Spawner**. For all elements of the grid we set a limitation of navigation area by container border.

```
1  using IEVO.UI.uGUIDirectedNavigation;
2  using UnityEngine;
3
4  namespace LimitedGrid_1
5  {
6      [RequireComponent(typeof(RectTransform))]
7      public class Spawner : MonoBehaviour
8      {
9          [SerializeField] private GameObject pfbBtnCell;
10         [SerializeField] private int spawnQuantity = 36;
11
12         private void Start()
13         {
14             RectTransform rectTransform = transform as RectTransform;
15
16             for (int i = 0; i < spawnQuantity; i++)
17             {
18                 GameObject go = Instantiate(pfbBtnCell, transform);
19                 DirectedNavigation dirNav = go.GetComponent<DirectedNavigation>();
20
21                 dirNav.ConfigDown.RectTransform.RectTransform = rectTransform;
22                 dirNav.ConfigUp.RectTransform.RectTransform = rectTransform;
23                 dirNav.ConfigLeft.RectTransform.RectTransform = rectTransform;
24                 dirNav.ConfigRight.RectTransform.RectTransform = rectTransform;
25             }
26         }
27     }
28 }
29
30 }
```

And that's all. Grid of elements with transitions only between items is ready.

## Option 2

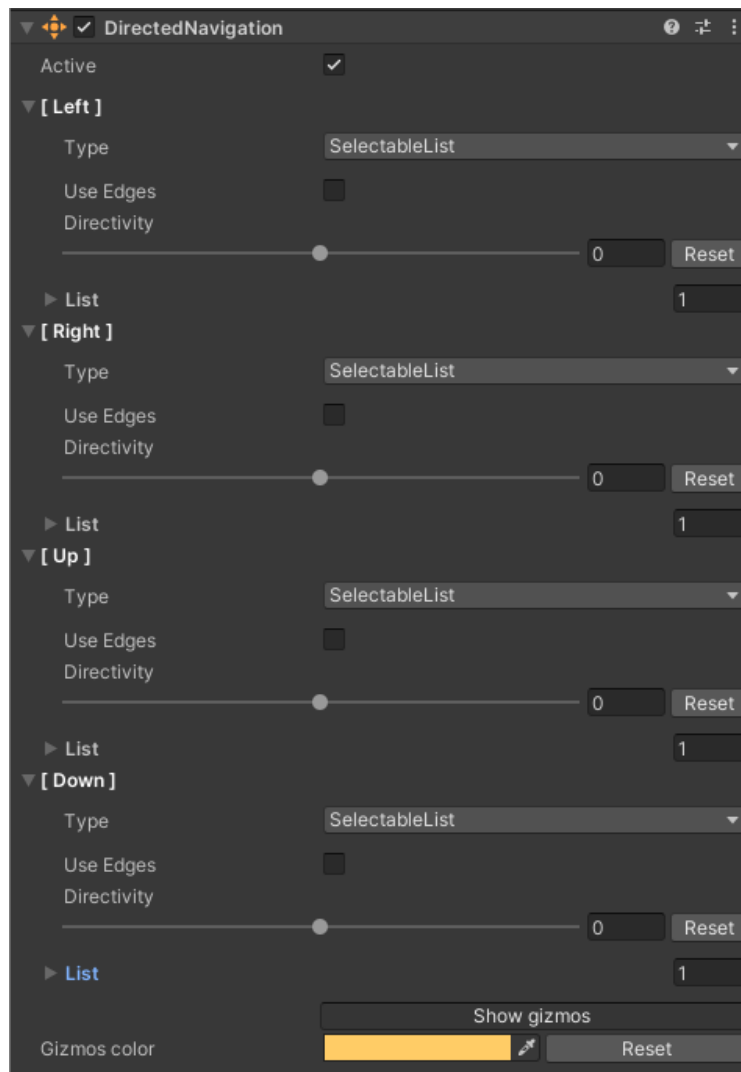
### Download link:

[https://ievo.games/uGUIDirectedNavigation/Downloads/Examples/LimitedGrid\\_2.unitypackage](https://ievo.games/uGUIDirectedNavigation/Downloads/Examples/LimitedGrid_2.unitypackage)

This option uses a [SelectableList](#) mode. It is a less preferable way for this case, but it also can be an option. May give you some ideas according to your needs.

Structure of elements is the same as in previous variant, differ only in grid item prefab settings and **Spawner** component.

In the grid item prefab, for all sides, set the **SelectableList** type.



**Spawner** in this option looks like this.

We create grid items, get their **Selectable** components as an **array**, and set this **array** for all **DirectedNavigation** components.

```
1  using IEVO.UI.uGUIDirectedNavigation;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  namespace LimitedGrid_2
7  {
8      public class Spawner : MonoBehaviour
9      {
10         [SerializeField] private GameObject pfbBtnCell;
11         [SerializeField] private int spawnQuantity = 30;
12
13
14         private void Start()
15         {
16             List<GameObject> btnCellsGOs = new List<GameObject>();
17             List<Selectable> btnCellsSelectables = new List<Selectable>();
18             List<DirectedNavigation> btnCellsDirNavs = new List<DirectedNavigation>();
19
20             for (int i = 0; i < spawnQuantity; i++)
21             {
22                 GameObject go = Instantiate(pfbBtnCell, transform);
23                 btnCellsGOs.Add(go);
24                 btnCellsSelectables.Add(go.GetComponent<Selectable>());
25                 btnCellsDirNavs.Add(go.GetComponent<DirectedNavigation>());
26             }
27
28             Selectable[] btnCellsSelectablesArr = btnCellsSelectables.ToArray();
29
30             for (int i = 0; i < btnCellsDirNavs.Count; i++)
31             {
32                 btnCellsDirNavs[i].ConfigDown.SelectableList.SelectableList = btnCellsSelectablesArr;
33                 btnCellsDirNavs[i].ConfigUp.SelectableList.SelectableList = btnCellsSelectablesArr;
34                 btnCellsDirNavs[i].ConfigLeft.SelectableList.SelectableList = btnCellsSelectablesArr;
35                 btnCellsDirNavs[i].ConfigRight.SelectableList.SelectableList = btnCellsSelectablesArr;
36             }
37         }
38     }
39 }
40
```



## Grid with Wrap Around

Wrap Around grid is a grid where you can move from the outermost elements to the farthest elements in the opposite direction. To build grids of this type with **Directed Navigation** you need to make some clicks.

**Important:** Note that for grids recommended usage of [DirectedNavigationGrid](#).

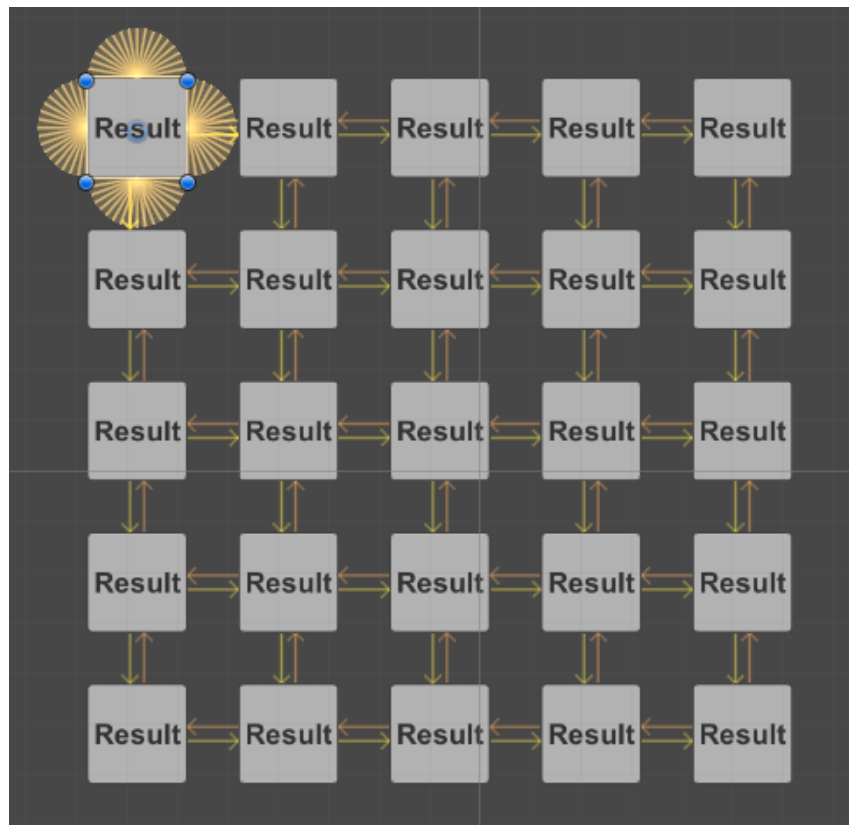
### Simple grid with Wrap Around

**Download link:**

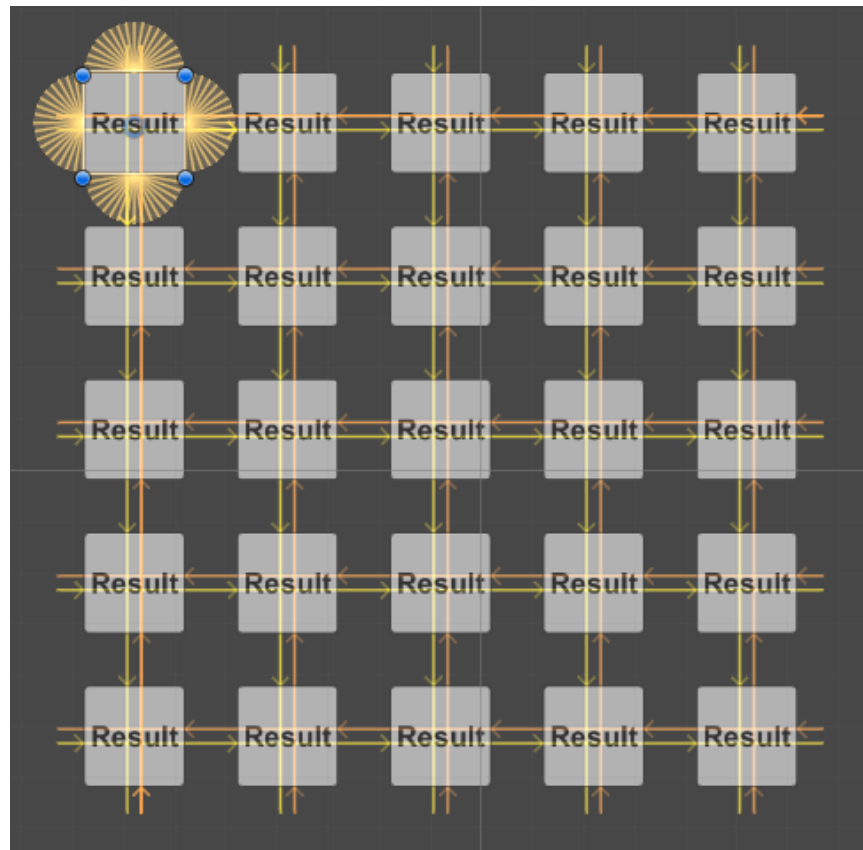
<https://ievo.games/uGUIDirectedNavigation/Downloads/Examples/GridWrapAround.unitypackage>

To get a grid with **Wrap Around** you just need for edge elements to enable this option for outermost elements.

Before



After



## Grid with Wrap Around and transition by lines

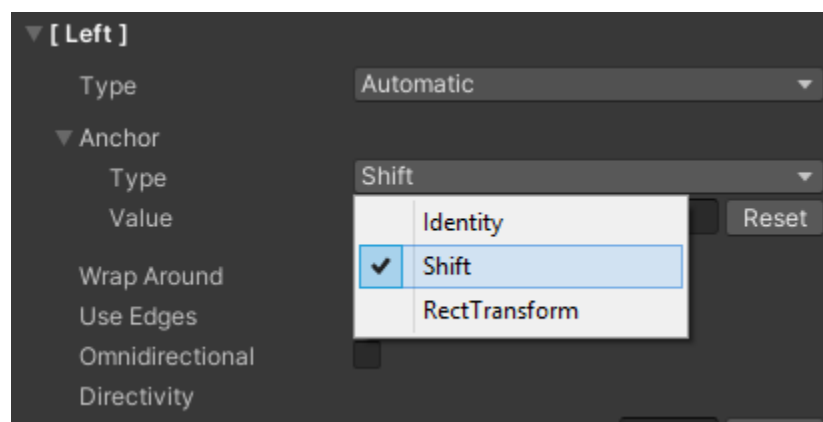
In this type of grid transition happens from the outermost elements to the farthest elements in the opposite direction with shift on one line up or down.

### Download link:

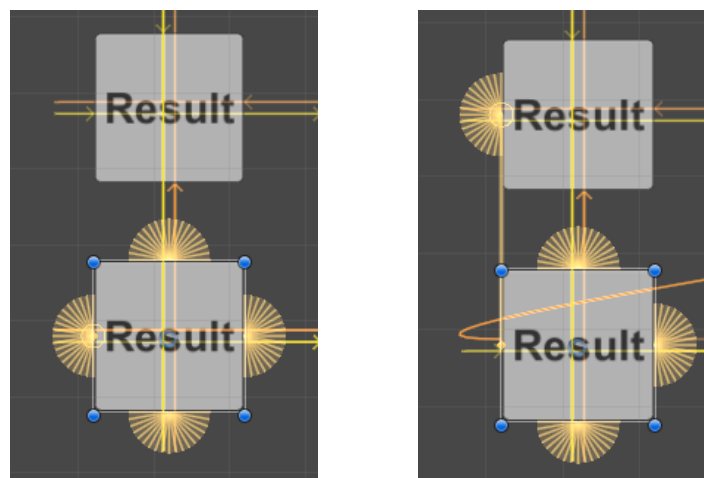
<https://ievo.games/uGUIDirectedNavigation/Downloads/Examples/GridWrapAroundByLines.unitypackage>

To get a grid with **Wrap Around** you need first to enable this option for outermost elements.

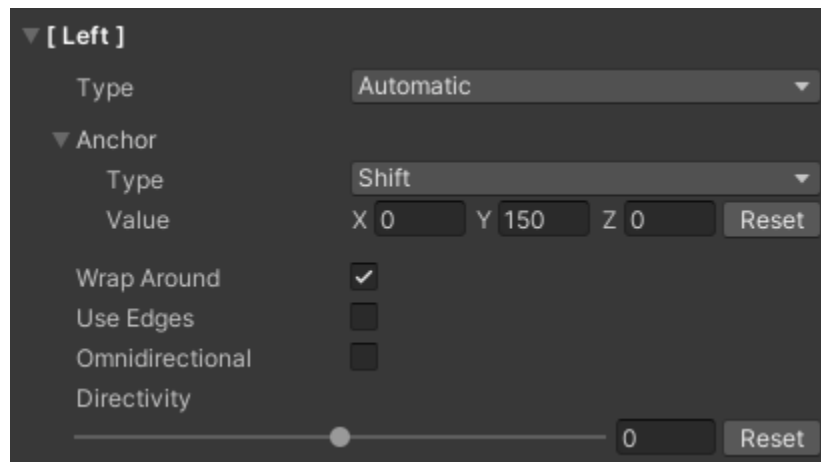
After need to activate **Advanced Settings** to show the **Anchor** option. In the **Anchor** settings select type **Shift**.



Move **Anchor** with the mouse to the upper line.



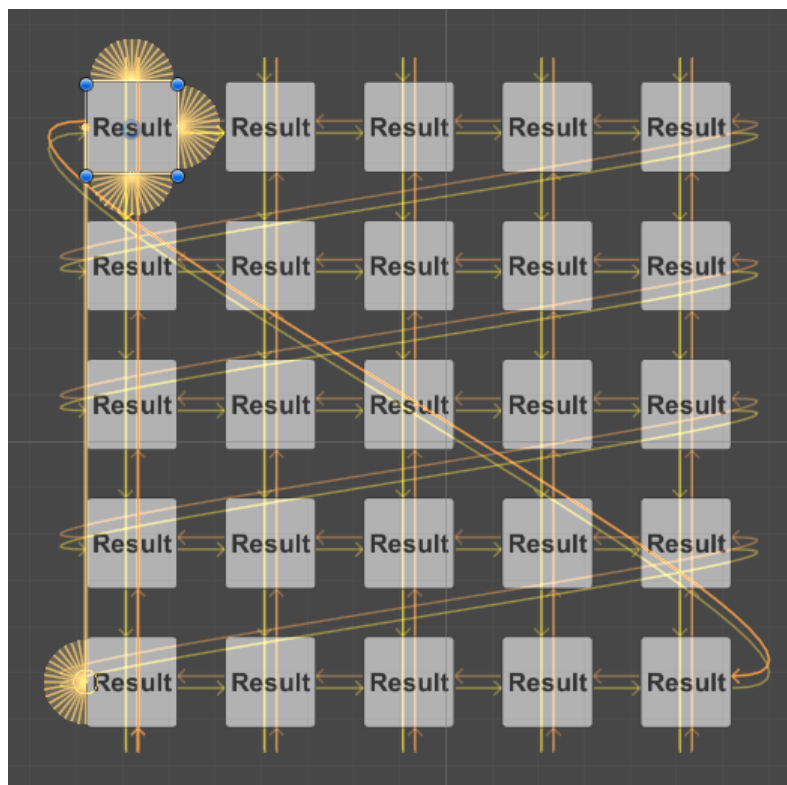
Or you can do it in the Inspector window by setting **y** value.



Also it can be done by code through **API**([Config.AnchorConfig](#)).

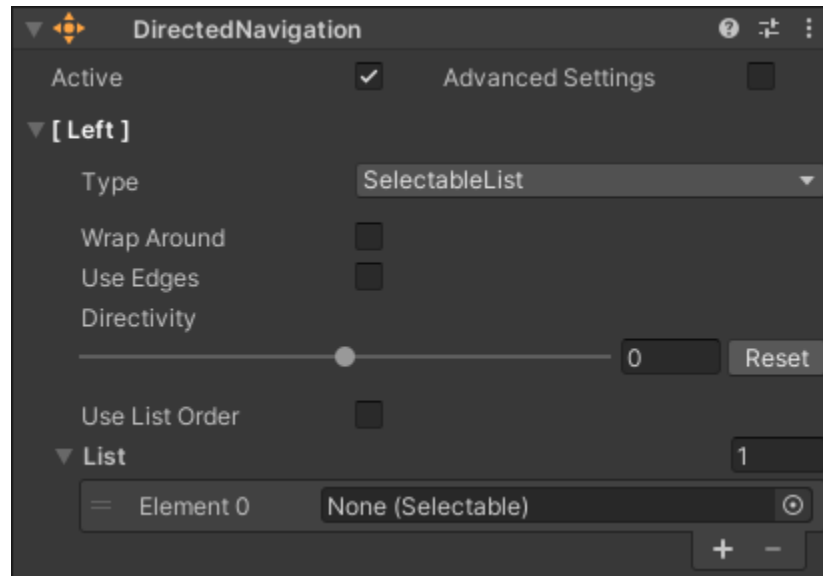
Repeat these actions for all outermost elements.

In result you will get a grid which looks like this.

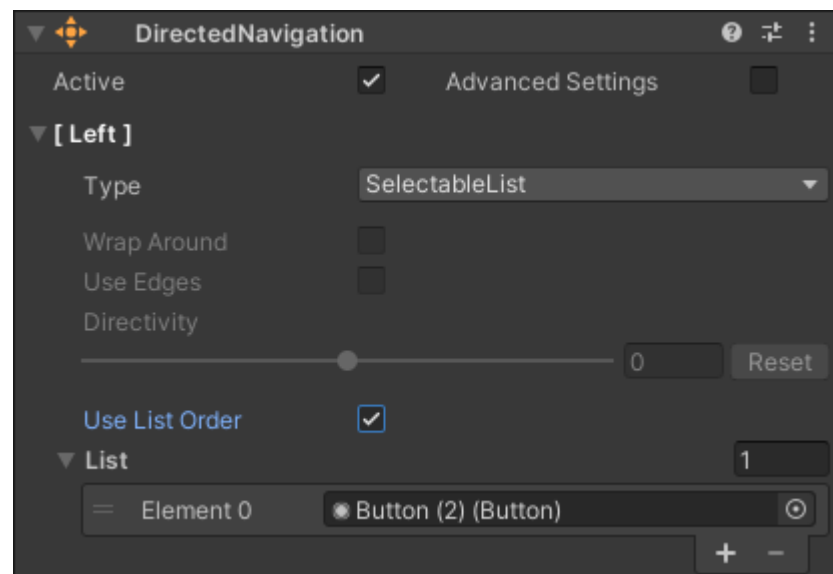


## Explicit

To make explicit navigation you can do it with mode SelectList.



Assign desired **Selectable** into list item. And **enable** the option **Use List Order**.



And that's it.

Instead of option **Use List Order** you can use the **Omnidirectional** option which is located in **Advanced Settings**.



But more preferable will be the first option.

## CONTACTS

Web site: <https://ievo.games/>

E-mail: [contact@ievo.games](mailto:contact@ievo.games) (subject: uGUI Directed Navigation)