



Kreli UI Management Kit

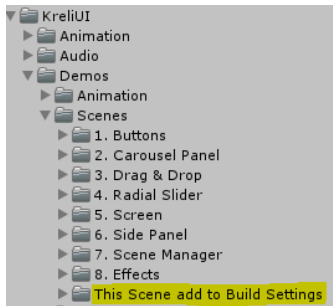
DOCUMENTATION
KRELI STUDIO

Contents

1. System Overview	2
1.1. Features	2
2. UI Elements Descriptions	3
2.1. UI Manager	3
2.2. UI Scene Manager	4
2.3. UI Screen	6
2.4. UI Button	7
2.5. UI Draggable & Drop Area	8
2.6. UI Radial Slider	9
2.7. UI Side Panel	10
2.8. UI Carousel Panel	11

More examples and tutorials available on 

1. System Overview



Kreli UI Management Kit is great UI system for games and applications created in Unity. It can be used on any platform you want and it's easy to implement because Kreli UI have customized inspector for easier use. You don't have to know any programming language to use it.

REMEMBER ADD SCENES FROM THE FOLDER "This Scenes add to Build Settings" TO BUILD SETTINGS (If you want to use examples).

The system includes such elements as:

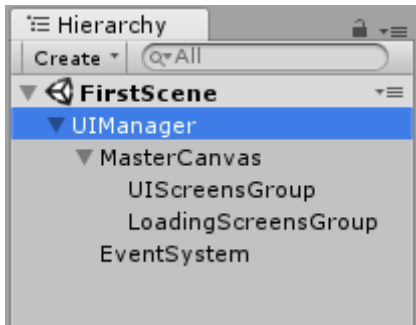
- UI Manager - a UI Screens management script that allows you to debug and preview their behaviors.
- UI Scene Manager - script for managing scenes and loading screens.
- UI Screen - script responsible for displaying UI groups. For example, popup window, HUD, Main menu of the game.
- UI Button – extended button with a larger number of interactions.
- UI Draggable & Drop Area – UI elements that can be dragged and drop to containers.
- UI Radial Slider – a circular slider with all functionalities of the standard slider and a few additional ones.
- UI Side Panel – side menu bar with very large adaptability to the application requirements.
- UI Carousel Panel – selection menu in the form of a horizontal or vertical carousel. Navigated by swipe with your finger or buttons.
- UI Effects – several effects adding dynamics to the UI system in the application.

1.1. Features

- Simple UI elements management system
- Intelligent scene management makes it easy to build large applications
- The screen loading management system between scenes will add professionalism to the application
- loading scenes asynchronously will allow you to load maps while playing without stopping it
- The innovative multi-layered nature of UI Screen will allow you to build an advanced menu with many windows at once
- History will allow you to go back to the previous UI Screen or to the previous scene
- Simple sound and vibration mechanism allows you to easily and quickly add sound effects and vibrations to your UI
- UI effects and UI animations will give dynamics and professionalism to the application
- Many examples and documentation in the package will quickly and easily explain to you the basics of the system

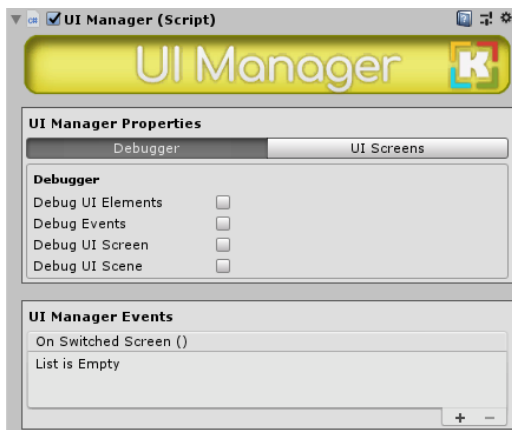
2. UI Elements Descriptions

2.1 UI Manager



UI Manager is the main script of which must be present on the scene to use other elements Kreli UI. It consists of two tabs of the first "Debugger" and "UI Screen". To create a UI Manager, select the "Create / KreliStudio / UI / UI Manager" menu in the hierarchi panel. After selecting this option, Ui Manager will be ready to use along with MasterCanvas, EventSystem and two groups for UIScreens and Loading Screens. Groups are for keeping order but are not required. All UI elements must be

inside the MasterCanvas object. In the case of creating any UI element in the scene where there is no UI Manager yet it will be created automatically and the selected UI element placed in it. UI Manager must be one for the entire project just like UI Scene Manager, otherwise complications may occur.

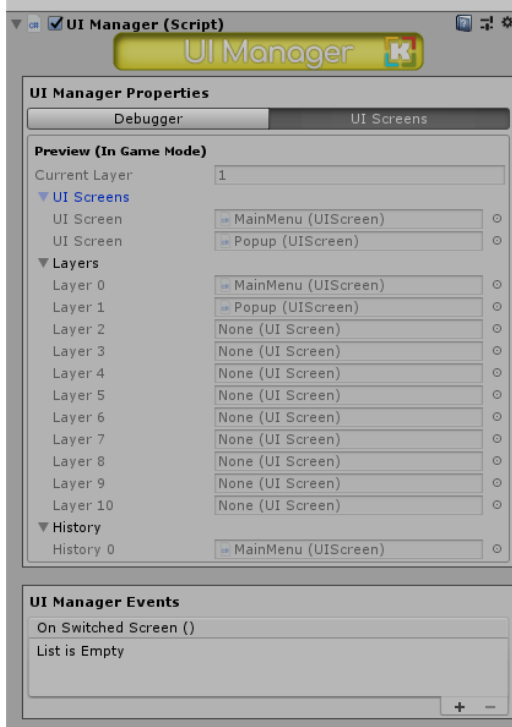


Debug UI Element – enables logs in such elements as UIButton, UICarouselPanel, UISidePanel, UIDraggable, UIDropArea, UIRadialSlider.

Debug Events – enables logs in all events of the Uli Kreli system.

Debug UI Screen – enables logs in operations related to UIScreen.

Debug UI Scene – enables logs in operations related to scenes.



Current Layer – the UI layer on which the current UIScreen is opened.

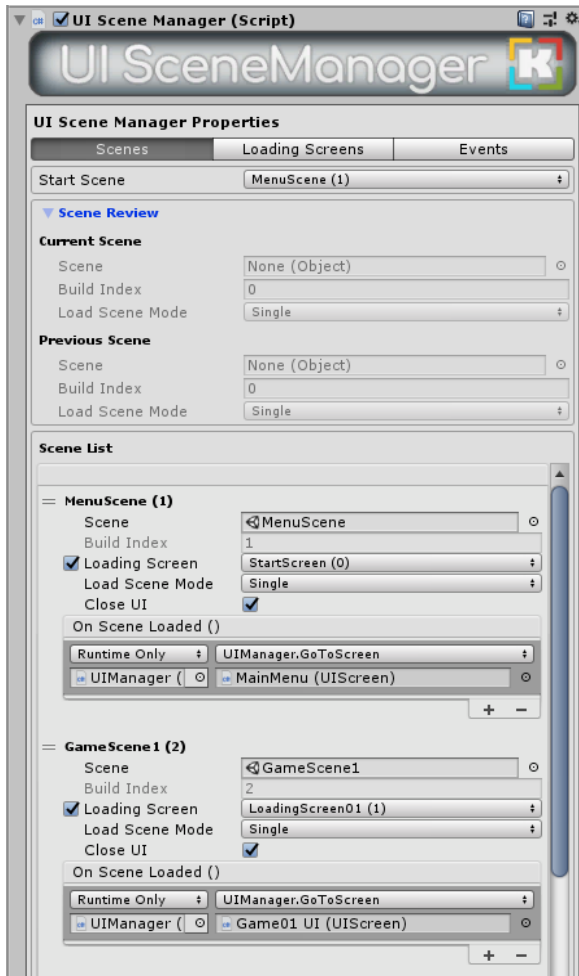
UI Screens – all UIScreens that exist in this scene.

Layers – All UI layers along with the currently displayed UIScreen on them.

History – the UIScreen history stack.

On Switched Screen – event called after changing each scene.

2.2 UI Scene Manager



The UI Scene Manager is located in the UI Manager object and is created with it using the "Create / KreliStudio / UI / UI Manager" path. This tool is used to manage scenes and load screens in a project. To work properly it is best to create a scene only with UI Manager and a camera displaying only the UI layer. Add this scene to Build Settings in the first place (build index = 0), so that it will be loaded first at the beginning of the application. Then, in the UI Scene Manager, set the "Start Scene" to the scene we want it to load at the start of the application. If this is not clearly understood, the package shows examples of how the UI Scene Manager should be configured.

Start Scene – scene that will be loaded right after starting the application.

Current Scene – Information about the currently loaded scene.

Previous Scene – previous scene to which you can return by calling the function `UISceneManager.LoadPreviousScene()`

Scene List – all scenes that will be used in the project should be added to the list of scenes.

Remember to add all scenes to Build Settings as well.

Scene – Drag here scene from project files to add scene to UI Scene Manager. This field is available in the editor, after compilation the project will be empty.

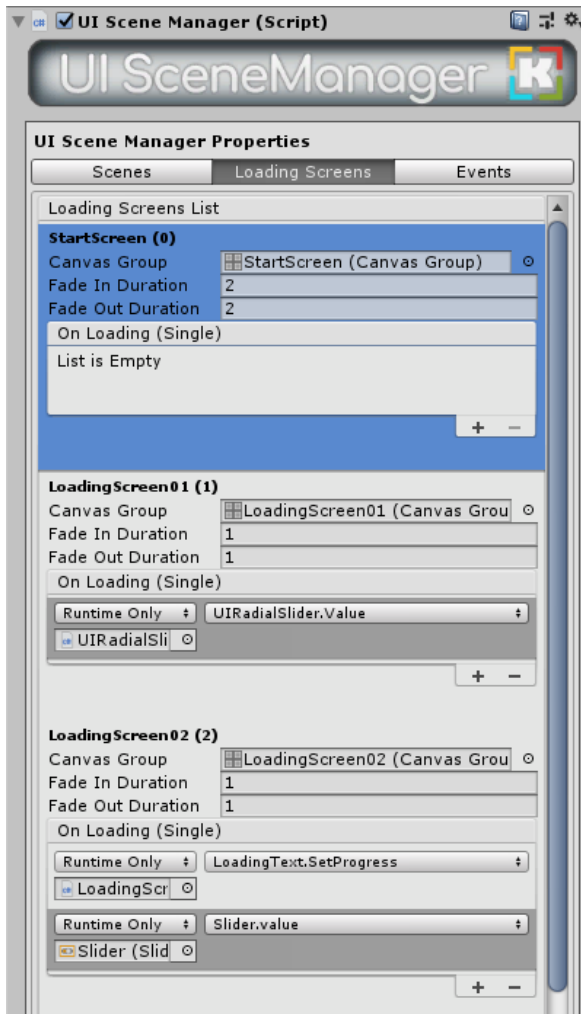
Build Index – this field is completed automatically after adding the scene to Build Settings. If you do not add a scene to Build Settings, it will display the message "Not added to build settings".

Loading Screen – If the loading screen is on, we select the available loading screens from the list for the given scene.

Load Scene Mode – Signal loads the scene and replaces it with the current one. Additive loads the scene and adds it to the hierarchy next to existing scenes.

Close UI – if checked during loading the next scene immediately closes all open UI Screens without the output animation.

On Scene Loaded – event invoked when loading a specific scene. Useful for displaying a specific UI Screen for a given scene.



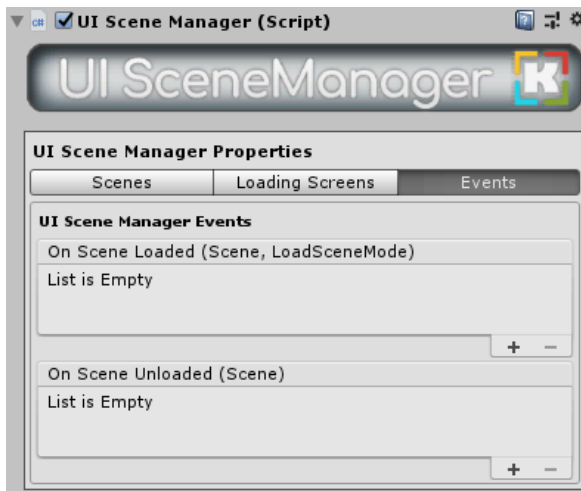
In the Loading Screens tab, you can add loading screens and set their parameters. Such loading screens are usually Panel for the entire screen with the added Canvas Group component and various elements inside, e.g. loading bar or text. In case we want to place the loading bar in our loading screen and to show the current progress of the loaded scene, add our loading bar in the event On Loading (float) and select "Dynamic float / Value". The package contains examples of how to create different loading screens.

Canvas Group – component from the main object of the loading screen.

Fade In Duration – fade in time of the loading screen.

Fade Out Duration – fade out time of the loading screen.

On Loading (float) – event called the scene while loading and forwarding parameter loading progress scene.

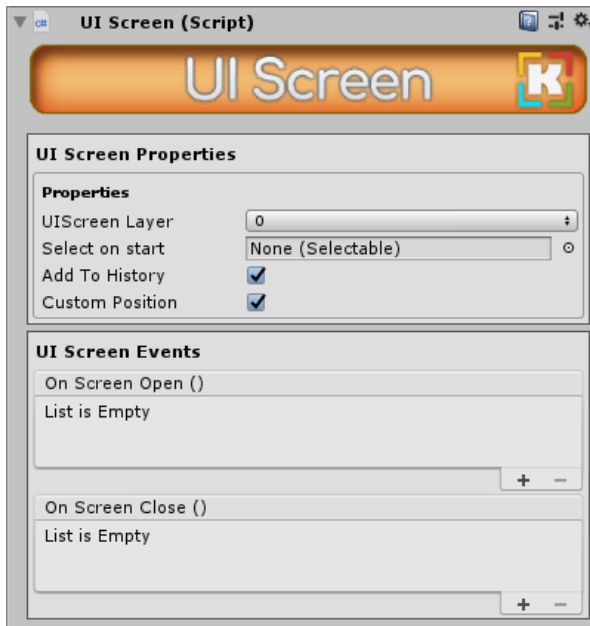


The last tab contains events called when any scene is loaded or unloaded.

On Scene Loaded (Scene, LoadSceneMode) – event called when each scene is loaded and forwarding scene parameters and loading type.

On Scene Unloaded (Scene) – event called when each scene is unloaded and forwarding the scene parameter.

2.3 UI Screen



Each UI screen consists of a UI Screen parent and UI elements inside it. With this solution you can create a complex, multi-layered interface and move between UI Screens. In the package, each example has at least one UI Screen in it. It is the basic element of UI Kreli. To create a UI Screen in the hierarchy, select from the menu "Create / KreliStudio / UI / UI Screen".

UIScreen Layer – UI Screen layer defining the positions in the display queue of a given UI Screen by Master Canvas. The bigger the number, the more UI Screen is on the front.

Select on start – set the highlight to a UI element inheriting from the "Selectable" class (buttons, text field, slider etc.) when UI Screen is opened.

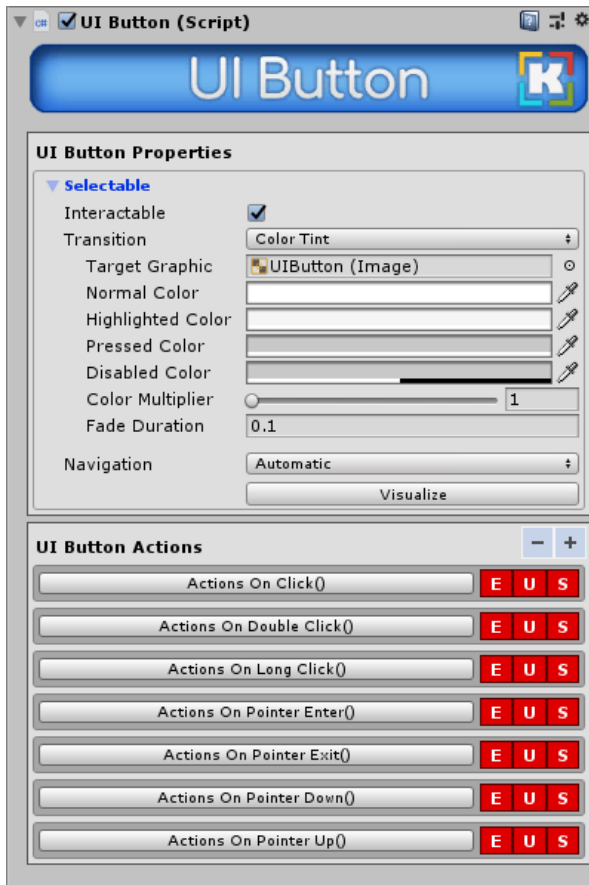
Add To History – if checked, UI Screen will be added to the history stack when switching between UI Screens.

Custom Position – if it is checked then it allows to place UI Screen on any coordinates in the editor. When switching to UI Screen it will be automatically moved to the zero Master Canvas position. This option allows you to keep order with a large amount of UI Screens in the editor.

On Screen Open – event called when opening a given UI Screen.

On Screen Close – event called when closing a given UI Screen.

2.4 UI Button

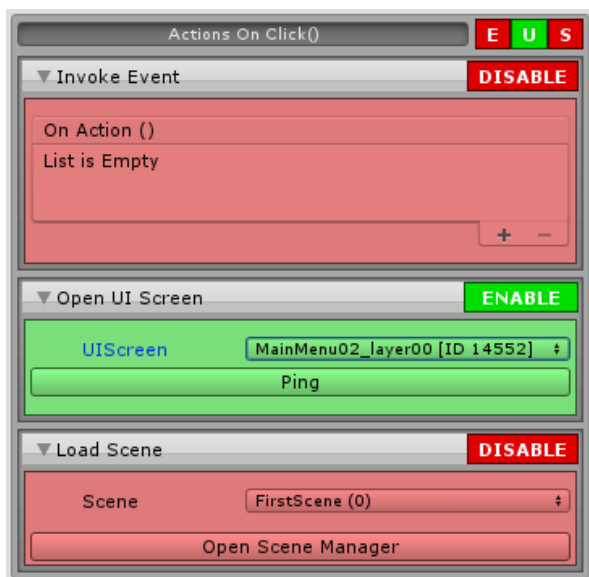


The button script is quite extensive but everything is organized. The UI Button Properties section contains fields inherited from the "Selectable" class. As you can see, the drop-down panel has the same fields as a regular button. The additional UI Button Actions section extends the functionality of the standard button. This section is divided into actions that can be performed with a button such as On Click, On Double Click etc. After expanding the action menu will display three types of actions that can be performed. Thanks to this we can call an event, open UI Screen and open a new scene. To create a UI Button in the hierarchy, select from the menu "Create / KreliStudio / UI / UI Button".

Selectable section – fields inherited from the "Selectable" class as in the standard button.

- + - the minus button collapses all action menus. The plus button expands all action menu.

E U S - icons informing about the state activation of action. E is events, U is UI Screen, S is Scene.



Actions buttons – buttons that open tabs with actions.

Enable/Disable button – enable or disable a given action.

Invoke Event -> **On Action** event that will be invoked when the action is detected.

Open UI Screen -> **UI Screen** – dropdown menu with a list of available UI Screen in Master Canvas and the option [Previous UI Screen]. After activating this action, the button will open the new UI Screen.

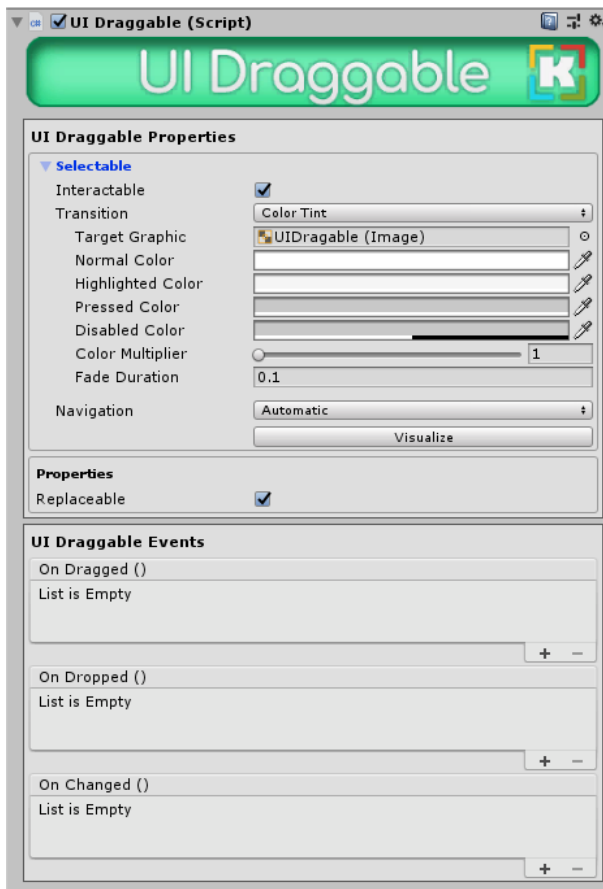
Open UI Screen -> **Ping** – highlights the selected UI Screen in the hierarchy and links the button

with the selected UI Screen in the editor. This option is used to check what will open the button.

Load Scene -> **Scene** – dropdown menu with a list of available scenes that are added in UI Scene Manager. After activating this action, the button will open a new scene.

Load Scene -> **Open Scene Manager** – opens the inspector UI Scene Manager.

2.5 UI Draggable & Drop Area



UI Draggable allows you to catch and move UI elements. Together with UI Drop Area, which is a container at UI Draggable, they create an easy to prepare and use Drag & Drop system from UI elements. To create a these elements in the hierarchy, select from the menu "Create / KreliStudio / UI / UI Draggable" and "Create / KreliStudio / UI / UI Drop Area".

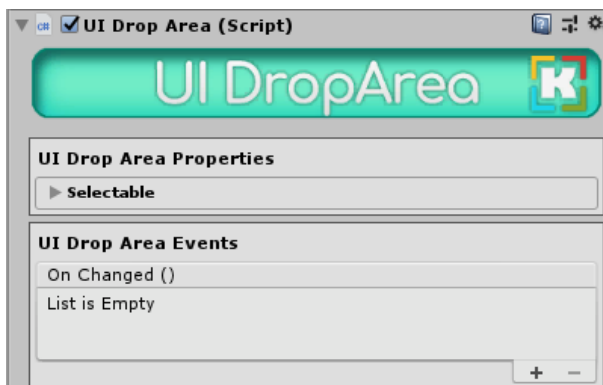
Selectable – fields inherited from the "Selectable" class as in the standard button.

Replaceable – if checked allows you to replace UI Draggable when another UI Draggable is dropped on the container in which it is located.

On Dragged – event invoked when caught.

On Dropped – event invoked when releasing.

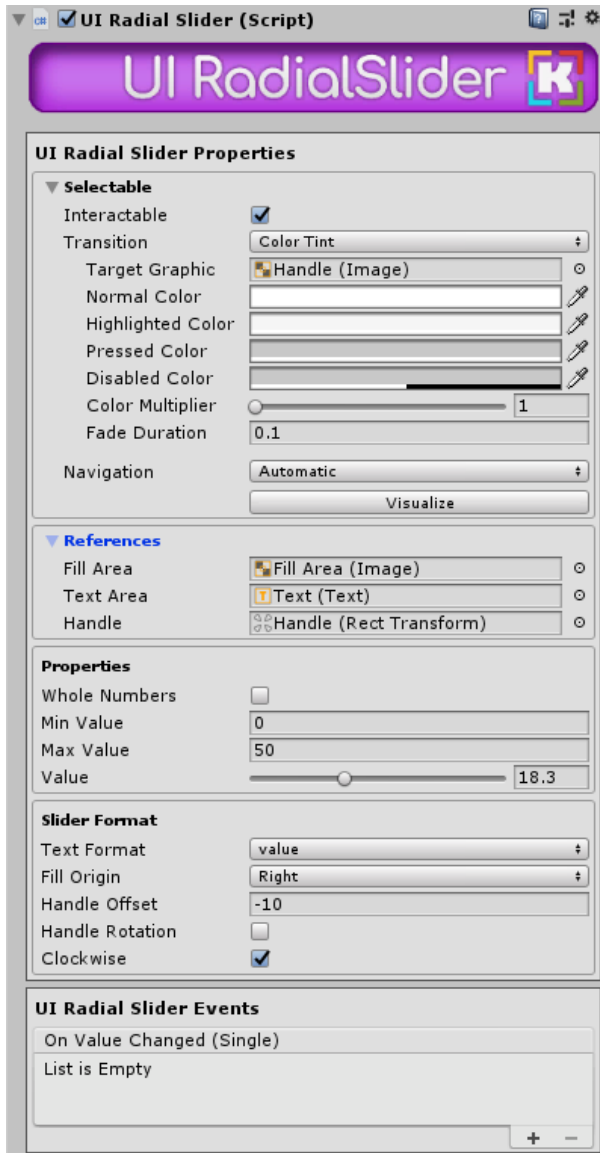
On Changed – event invoked when changing position. For example, after dropping on a new container.



Selectable – fields inherited from the "Selectable" class as in the standard button.

On Changed – event invoked when changing the Drop Area UI content. For example, after dropping a new UI Draggable to him.

2.6 UI Radial Slider



UI Radial Slider behaves like a standard slider but is shaped like a circle. To create a UI Radial Slider in the hierarchy, select from the menu "Create / KreliStudio / UI / UI Radial Slider".

Selectable – fields inherited from the "Selectable" class as in the standard slider.

References – several fields with references to slider objects such as fill area, text area, handle.

Whole Numbers – If checked, the value of the slider is integers.

Min Value – the minimum value a slider can point to.

Max Value – the maximum value a slider can point to.

Value – the current value indicated by the slider.

Text Format – Value display format. Available formats:

- value
- value / max_value
- percent

Fill Origin – determining the starting position of the slider.

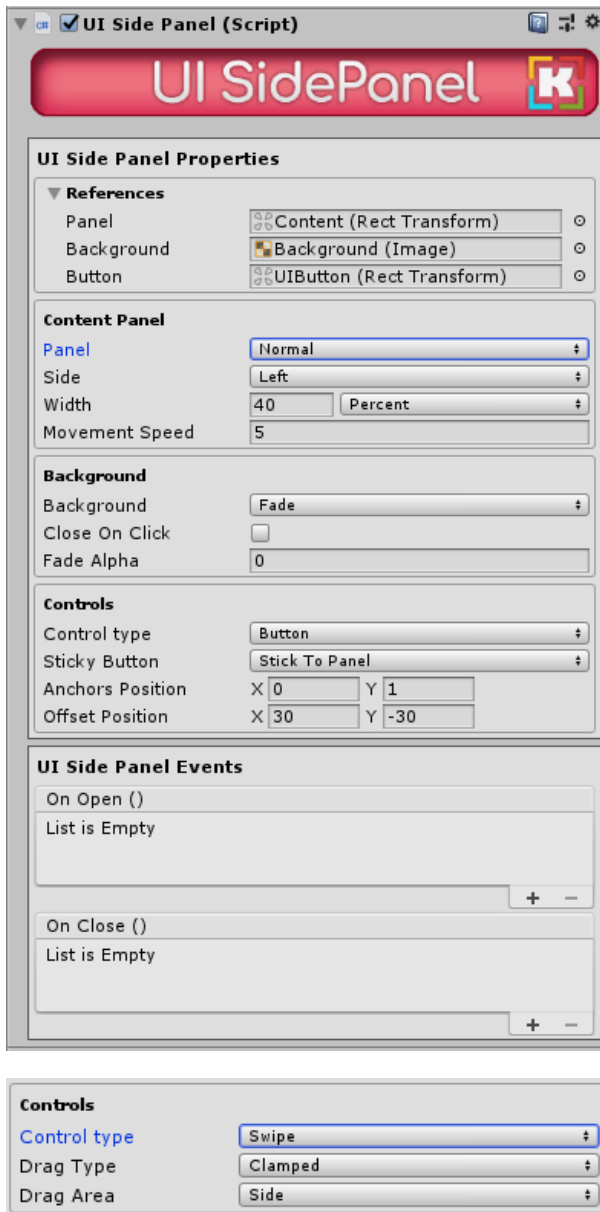
Handle Offset – distance of the handle from the center of the slider.

Handle Rotation – if checked, the handle is always rotated inside the slider. For example, like clock tips.

Clockwise – change direction slider.

On Value Changed (float) – event invoked when changing the value of the slider, giving the new value as an parameter.

2.7 UI Side Panel



Side panel is a frequently used UI element in various applications, but in Unity it is rare. Now there will be no problem to use it in your application. The panel has quite a lot of customization options. You can open it with a button or finger swipe, when the panel is open it can cover the screen with background, it can be attached to each side of the screen. To create a UI Side Panel in the hierarchy, select from the menu "Create / KreliStudio / UI / UI Side Panel".

References – several fields with references to side panel objects such as content panel, background, button, swipe area.

Panel – type of side panel opening and closing animation. Available normal or fade.

Side – the edge of the screen on which the side panel is to be located.

Width – side panel width. Possibility to enter values in pixels or in percent of the width of the screen.

Movement Speed – the speed of opening and closing the side panel.

Background – type of background.

Close On Click – If checked, when you click on the background then side panel is closed.

Fade Alpha – level of background transparency.

Control Type – the choice between the button

and the swipe area to control the side panel.

Sticky Button – if checked, the button will move with the side panel when opening and closing.

Anchors Position – button anchors position.

Offset Position – button offset position from anchors position.

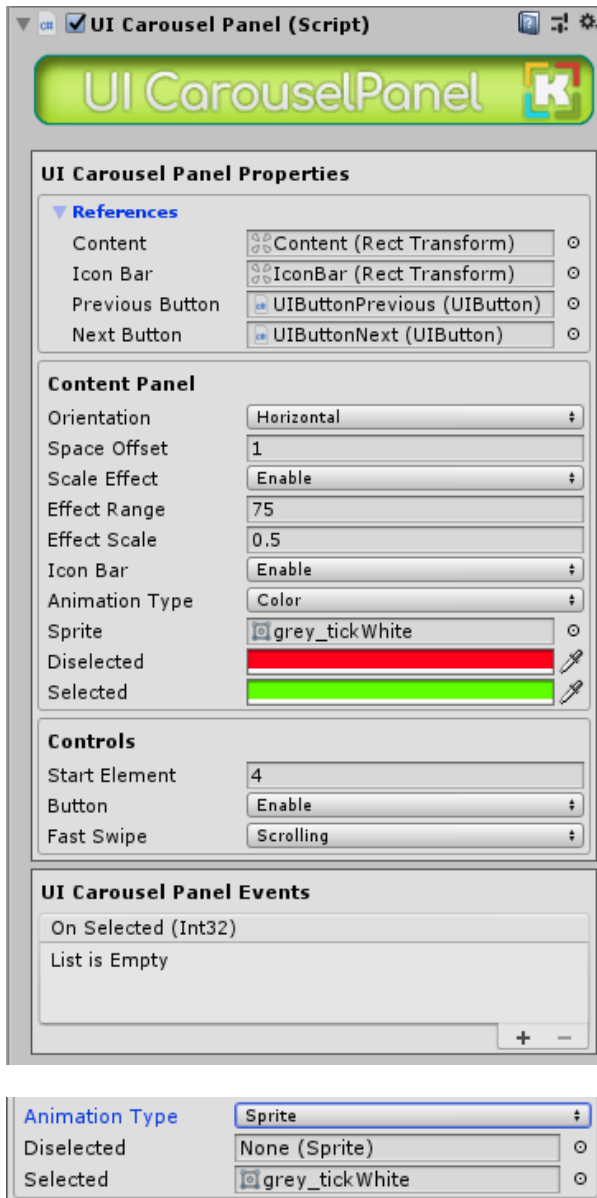
Drag Type – works similar to Scroll Rect, clamped or elastic to choose from. If elastic is selected while dragging the swipe area, the panel is not attached to the edge but follows the finger.

Drag Area – determines the area of the finger swipe action on the side panel. To choose a "side" area close to the edge or "full screen" area of the entire screen.

On Open – event invoked when the side panel is opened.

On Close – event invoked when the side panel is closed.

2.8 UI Carousel Panel



Carousel panel is more like the side panel is a very often used UI element but it is not available in the standard UI kit in Unity. The perfect way to make a choice of the hero or levels in the game. To create a UI Carousel Panel in the hierarchy, select from the menu "Create / KreliStudio / UI / UI Carousel Panel".

References – several fields with references to the carousel panel objects such as content panel, icon bar, buttons.

Orientation – the direction of the carousel.

Space Offset – distance between elements of the content panel.

Scale Effect – the effect of zoom the selected element which is located in the center of the carousel panel.

Effect Range – area of the effect.

Effect Scale – the scale of the zoom effect.

Icon Bar – if checked will be visible bar with icons in carousel panel and highlighted currently selected element.

Sprite – displayed icon.

Color -> Deselected – icon color of the unselected element.

Color -> Selected – icon color of the selected element.

Sprite -> Deselected – sprite icon of unselected element.

Sprite -> Selected – sprite icon of selected element.

Start Element – initially selected element.

Button – if selected, the "Next" and "Previous" buttons will be visible in the carousel for control.

Fast Swipe – if the selected "Scrolling" fast swipe will be rotated carousel panel. If the selected "Move By One" fast swipe moves the carousel panel by one element.