

FOR UNITY3D

- Documentation -

Version number	Date	Description
1.0	05/04/2013	Initial version.
1.1	18/05/2013	<p>WARNING! New buttons events system. If you update from v1.0 to v1.1, you will lose your buttons' 'OnClick' events.</p> <p>Add functionalities :</p> <ul style="list-style-type: none"> • Menu position choice (Screen percentage or Screen raycast) • No upside down category name toggle • Active buttons during transitions toggle • New buttons events system with OnClick and OnRelease <p>Add functions:</p> <ul style="list-style-type: none"> • Get category in menu by name • Get button in menu by name • Get button in category by name • Show and Hide functions can be used without playing transitions if 'true' is sent. <p>Minor bugs fixed.</p>

I)	Create a basic circular menu	4
II)	Zoom in circular menu components	4
1)	Categories.....	4
0 :	Remove button.....	4
1 :	Category's name.....	4
2 :	Show category name toggle	4
3 :	Label size.....	4
4 :	Title font	5
5 :	Text alignment.....	5
6 :	Text size	5
7 :	Text style.....	5
8 :	Text color	5
9 :	Buttons list.....	5
10 :	'Add Button' button.....	5
2)	Buttons	5
1 :	Button's name	5
2 :	On click element to call	5
3 :	On click function called.....	6
4 :	On click parameter type	6
5 :	On click parameter	6
6 :	On release element to call.....	6
7 :	Normal texture	6
8 :	Hover texture.....	6
9 :	Clicked texture.....	6
10 :	Clear button.....	6
11 :	Remove button.....	6
3)	Global parameters.....	7
1 :	Start active.....	7
2 :	Menu position type	7
3 :	Manually set radius	7
4 :	Refresh radius.....	7
5 :	Button size	7
6 :	Menu initial rotation	7

7 : Menu open angle.....	7
8 : Category's name radius	8
9 : Use activation menu.....	8
10 : No upside-down category name	8
11 : Use central button.....	8
12 : Central button hide menu	8
4) Transitions	8
1 : Use transitions.....	8
2 : Active buttons click during transitions	8
3 : Transition duration	8
4 : Radius transition.....	9
5 : Size transition	9
6 : Fan transition.....	9
7 : Fade transition.....	9
8 : Rotation transition.....	9
9 : Show separators	9
5) Menu position	9
III) Classes description	10
1) SendMessageParameters.....	10
a) Parameters	10
b) Functions	10
2) CircularMenuButton.....	10
c) Parameters	10
d) Functions	11
3) CircularMenuCategory	12
a) Parameters	12
b) Functions	12
4) CircularMenu.....	13
a) Parameters	13
b) Functions	15

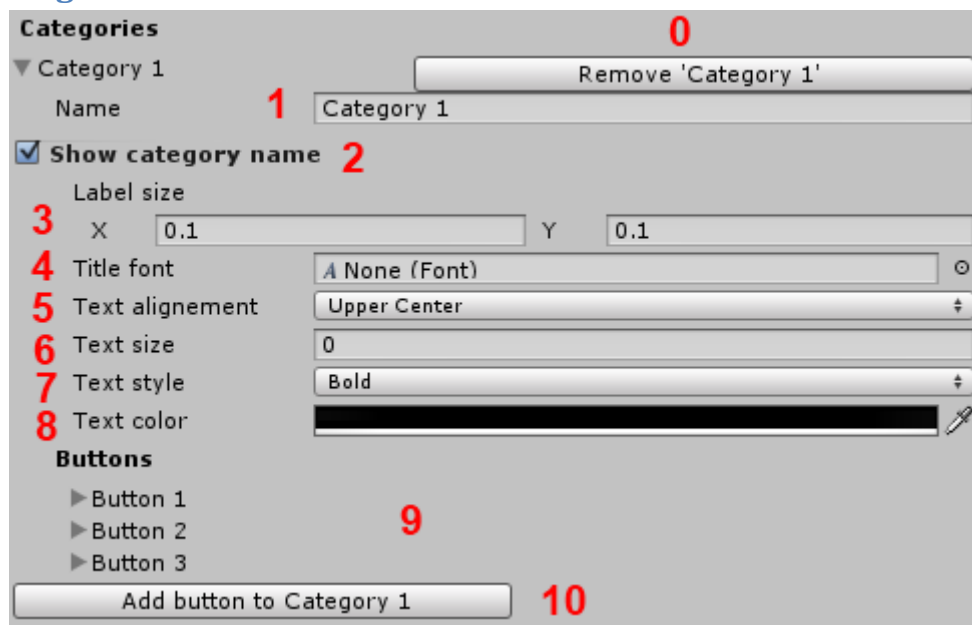
I) Create a basic circular menu

To add a basic circular menu to your scene, follow these steps:

- Create an empty GameObject
- Move this object between 0 and 1 on X and Y axis like a GUITexture
- Add the CircularMenu component
- Add any number of categories
- Add any number of buttons to your categories (if you add too many buttons, the unity editor will have difficulty to refresh himself).
- For each button, select the texture you want
- Press play
- Press the Show / Hide editor button to show or hide the circular menu.

II) Zoom in circular menu components

1) Categories



0 : Remove button

This button removes the actual category from your menu. Take care, it deletes all category's buttons.

1 : Category's name

It is the name of the current category.

2 : Show category name toggle

This Boolean let you choose if you want to show or not the category name on your menu.

3 : Label size

This is the size of the category name label. It is proportional to your menu size. Set its value between 0 and 1 (0 to 100%).

4 : Title font

Category's name label font.

5 : Text alignment

Category's name label alignment.

6 : Text size

Category's name label text size.

7 : Text style

Text style of the category's name label.

8 : Text color

Text color of the category's name label.

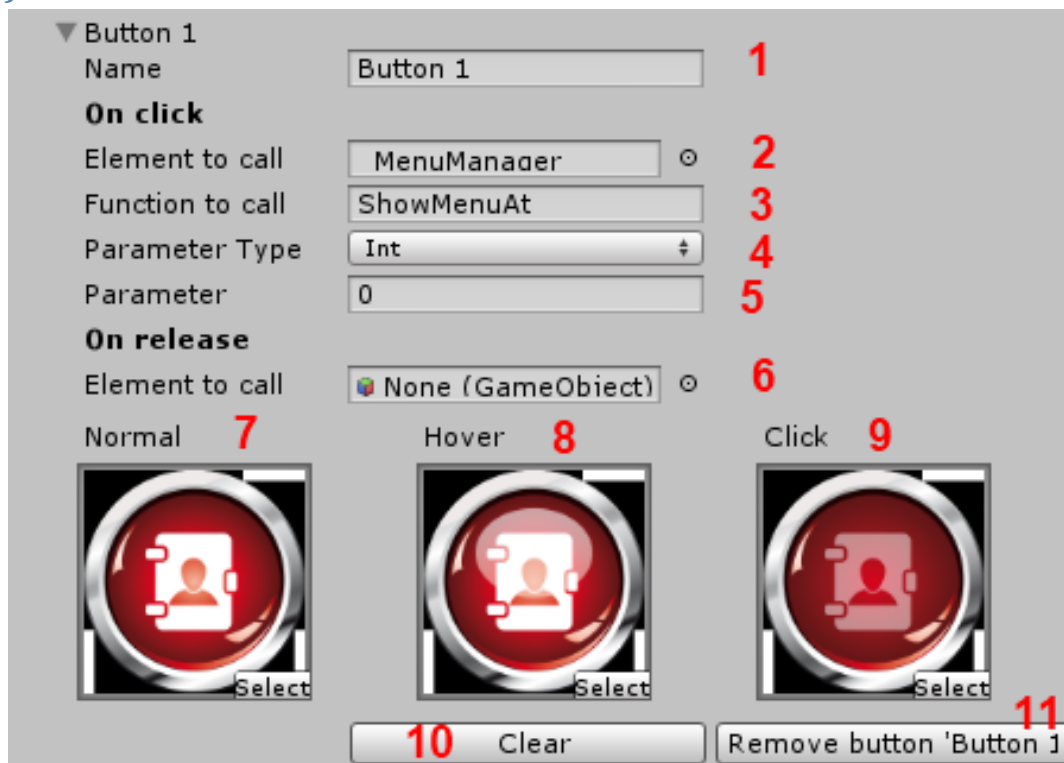
9 : Buttons list

Buttons list of the current category.

10 : 'Add Button' button

Button used to add a button to the current category.

2) Buttons



1 : Button's name

It is the name of the current button.

2 : On click element to call

GameObject called with 'SendMessage()' when the user press this button.

3 : On click function called

Function called with 'SendMessage()' when the user press this button.

4 : On click parameter type

If the called function requires a parameter, declare its type.

5 : On click parameter

If the called function requires a parameter, set it.

6 : On release element to call

GameObject called with 'SendMessage()' when the user release this button.

(Function to call and parameters will appear when you drag & drop a gameobject).

7 : Normal texture

Button's texture showed at normal time.

8 : Hover texture

Button's texture showed when user's cursor is over the button.

9 : Clicked texture

Button's texture showed when the user clicks on the button.

10 : Clear button

Button used to clear all the button's parameters.

11 : Remove button

Button used to remove actual button from its category.

3) Global parameters

Global parameters

Start active ☒ 1

Menu position type 2 Screen Percentage

Manually set radius 3 ☒ 4 Refresh radius

Radius 5 0.09

Button size (% of screen width [0-1]) 0.05 6

Menu initial rotation 7 0

Menu open angle 8 -90

Category name radius 9 1.2

No upsidedown category name 10 ☒

☒ Use activation button 11
▶ Button

☒ Use central button 12
Central button hide menu ☒ 13
▶ Button

1 : Start active

Does the menu start active?

2 : Menu position type

Screen percentage: Menu screen position is set from menu transform position in screen percent (from 0 to 1).

Screen Raycast : Menu screen position is set from menu World to screen point position on the current 'Main Camera'.

3 : Manually set radius

Does the menu auto calculates its radius or does it uses a manually set radius?

'Manually set radius' is a percentage of the screen width from 0 to 1. (0 to 100%).

4 : Refresh radius

Force refresh the menu radius.

5 : Button size

Global size of the menu's buttons.

Express in percentage of the screen width from 0 to 1. (0 to 100%).

6 : Menu initial rotation

Start angle of the menu. Change it to move the first button's position on the menu's circle.

7 : Menu open angle

Total angle used by the menu to show its buttons. 360° for a full circle, 180° for an half circle.

8 : Category's name radius

Position of the categories' names of the menu. (Only for menus using 'Show category name')..

9 : Use activation menu

Does your menu use an activation button to show itself when it is inactive?

10 : No upside-down category name

Prevent categories' name to be upside-down.

11 : Use central button

Does your menu show a central button?

12 : Central button hide menu

Does this central button hide the menu when it is pressed?

4) Transitions

The screenshot shows a configuration window for menu transitions. It includes several checkboxes and input fields. Red numbers 1 through 9 are overlaid on the image to identify specific settings:

- 1: ☒ Use transitions
- 2: ☐ Active buttons click during transitions
- 3: Transition duration (input field with value 1)
- 4: ☒ Radius transition
- 5: ☒ Size transition
- 6: ☐ Fan transition
- 7: ☒ Fade transition
- 8: ☒ Rotation transition
- 9: Separator size (% menu radius) (X: 0.8, Y: 0.05)

Other visible settings include Alpha min (0), Alpha max (1), and Angle rotation (360). A 'Select' button is located at the bottom right of the window.

1 : Use transitions

Does the menu use transitions when it appears or disappears?

2 : Active buttons click during transitions

Can the user click on buttons during transitions?

3 : Transition duration

Duration of the menu transitions (in seconds).

4 : Radius transition

Does transition use radius length modification?

5 : Size transition

Does transition use buttons' size modification?

6 : Fan transition

Does transition use fan animation?

7 : Fade transition

Does transition use fade in from 'Alpha min' to 'Alpha max' and fade out from 'Alpha max' to 'Alpha min'.

8 : Rotation transition

Does transition use rotation animation, rotating 'Angle rotation' degrees?

9 : Show separators

Does your menu show separators between your categories?

Set the separator's size and texture.

5) Menu position

To change your circular menu's position, use the Transform panel and move X and Y axes between 0 and 1 like a GUITexture.

Position is set in percentage of the screen's width and height.

III) Classes description

1) SendMessageParameters

a) Parameters

Accessibility	Type	Name	Description
public	enum	UsableParameterType	Parameters type allowed when using 'elementToCall', 'functionToCall', 'parameter'.
public	GameObject	elementToCall	GameObject called on use
public	string	functionToCall	Function called on the 'elementToCall' GameObject on use
public	UsableParameterType	paramType	Type of the current 'functionToCall' parameter.
public public public public	int float bool string	paramInt paramFloat paramBool paramString	Parameter sends to the function when needed. Depend of the 'paramType' selected.

b) Functions

Accessi bility	Return Type	Name	Parameters	Description
public	void	Use	string	This function call the 'SendMessage()' function on the 'elementToCall' GameObject using the 'fonctionToCall' function and selected parameter.

2) CircularMenuButton

c) Parameters

Accessibility	Type	Name	Description
private	CircularMenuCategory	parentCategory	Parent category of the button.
public	string	name	The name of the button.
public	GUIStyle	buttonStyle	Button's style where textures linked.
public	SendMessageParameters	OnClickMessage	SendMessageParameters used when the button is clicked.
public	SendMessageParameters	OnReleaseMessage	SendMessageParameters used when the button is released.
public	Rect	viewRectangle	Button position.
public	bool	isClicked	Is the button actually clicked.

public	int	fingerId	ID of the last Finger which click on the button.
public	bool	showInEditor	Is this button show in the UnityEditor? Only used in the UnityEditor.

d) Functions

Accessi bility	Return Type	Name	Parameters	Description
public	void	Clear	None	This function clears all the button's parameters to their default value.
public	void	OnClick	int	This function is called when the button is clicked and Use the OnClickMessage if it is not empty.
public	void	OnRelease	int	This function is called when the button is release and Use the OnReleaseMessage if it is not empty and the fingerId is the same than the last fingerId which active the button.

3) CircularMenuCategory

a) Parameters

Accessibility	Type	Name	Description
public	string	name	The name of the category.
public	bool	showCategoryName	Is the category's name show in the menu?
public	GUIStyle	textStyle	Label's style in which the category's name is written.
public	Vector2	textLabelSize	Size of the label in which the category's name is written. Use variables between 0 and 1 representing a percent of the menu's size.
public	List<CircularMenuButton>	buttonList	List of all the 'CircularMenuButton' in this category.
public	bool	showInEditor	Is this category show in the UnityEditor? Only used in the UnityEditor.

b) Functions

Accessibility	Return Type	Name	Parameters	Description
public	CircularMenuButton	AddButton	None	This function adds a button to the current category and returns the new created button.
public	void	RemoveButton	CircularMenuButton	This function removes a specific button from the category.
public	CircularMenuButton	GetButtonFromName	string	Return a button with a similar name contains in this category. Return null if there is no button with this name.

4) CircularMenu

a) Parameters

Accessibility	Type	Name	Description
public	enum	PositionType	ScreenPercentage, ScreenRaycast
public	PositionType	usedPositionType	Actual menu position type
private	Vector3	actualMenuPosition	Actual menu position on screen
public	List<CircularMenuCategory>	menuCategories	List of all the menu's categories.
public	bool	startActive	Does the menu active itself at start?
public	bool	manuallySetRadius	Does the menu auto calculate its radius depending of the number of buttons and the buttons size? Or do you set manually its radius?
public	float	circleRadius	Radius of the circular menu. Is runtime calculate if 'manuallySetRadius' is true.
public	float	buttonSize	Size of the menu's buttons. Percent of the screen width (0 to 1).
public	float	categoryNameRadius	Radius in which the categories name label will be draw.
public	bool	categoryNameNoUpsidedown	Prevent categories' name to be upside-down.
public	float	startMenuAngle	Start rotation of the menu if you want to change the button's position on the menu's circle.
public	float	menuMaxOpenAngle	Angle in which all menu's buttons will be draw. (180 for an half circle, 360 for a full circle...)
public	bool	showCentralButton	Is the central menu button shown?
public	bool	centralButtonHideMenu	Does the central menu button hide the menu when clicked?
public	CircularMenuButton	centralButton	Characteristics of the central button.
public	bool	showActivationButton	Is the activation button shown when the menu is hidden?
public	CircularMenuButton	activationButton	Characteristics of the activation button.
public	bool	useTransitions	Does the menu use transition animation when he is shown or hidden?
Public	Bool	useButtonsDuringTransition	Could buttons be used during transitions?
public	float	transitionDuration	The duration of menu's transitions.
public	bool	radiusTransition	Does the transition play a radius scale animation?

public	bool	sizeTransition	Does the transition play a button scale animation?
public	bool	fanTransition	Does the transition play a fan animation?
public	bool	fadeTransition	Does the transition play a fade animation?
public	float	minAlpha	Minimum alpha of the fade transition.
public	float	maxAlpha	Maximum alpha of the fade transition.
public	bool	rotationTransition	Does the transition play a rotation animation?
public	float	maxRotation	Angle rotation of the rotation animation.
public	bool	showSeparators	Are categories separators shown?
public	GUIStyle	separatorStyle	Style of the separators. Change background normal texture to separator's visual.
public	Vector2	separatorSize	Size of all the separators. Percent of the menu size (0-1).
public	enum	EtatMenu	Enum of all the menu states.
private	EtatMenu	actualState	Current state of the menu.
private	bool	menuIsActive	Is the menu active?
private	int	totalElementNumber	Total number of buttons in the menu.
private	float	actualAlpha	Actual alpha of the menu.
private	float	actualCircleRadius	Actual radius of the menu.
private	float	startTime	Start time of a transition animation.
private	float	actualRotAngle	Actual rotation of the menu.
private	float	actualBtnSize	Actual menu's button size.
private	float	actualOpenAngle	Actual open angle of the menu.
private	float	lockedAlpha	Temporary variable used when a fadeout transition is call during a fade in transition.
private	float	lockedRotAngle	Temporary variable used when a fadeout transition is call during a fade in transition.
private	float	lockedBtnSize	Temporary variable used when a fadeout transition is call during a fade in transition.

private	float	lockedRadius	Temporary variable used when a fadeout transition is call during a fade in transition.
private	float	lockedOpenAngle	Temporary variable used when a fadeout transition is call during a fade in transition.
public	delegate	void DefaultDelegate	Delegate declaration.
public	DefaultDelegate	ActivatedCallBack	Callback called when the fade in transition end.
public	DefaultDelegate	DesactivatedCallBac k	Callback called when the fade out transition end.

b) Functions

Accessib ility	Return Type	Name	Parameters	Description
public	void	ShowMenu	None	Show the circular menu. If menu use transitions, start the fade in transition.
public	void	HideMenu	None	Hide the circular menu. If menu use transitions, start the fade out transition.
private	void	StabiliseMenu	None	Set the menu to a static actived state.
private	void	DesactivateMenu	None	Set the menu to a static deactivated state.
private	void	UpdateMenuPosition	None	Update the menu position on screen depending of the selected usedPositionType.
private	void	UpdateAction	None	Update the player click or touch on buttons.
private	void	UpdateTransitions	None	Update menu positions in transitions.
private	void	DrawButton	CircularMenuBut ton actualBtn float angleActuel	Draw a specific button at a specific menu's angle.
private	void	DrawSeparator	float angle	Draw a menu's separator at the specific menu's angle.
private	void	DrawCategoryName	CircularMenuCat egory part float angleDebut float angleFin	Draw specific category's name starting at 'angleDebut' menu's angle and ended at 'angleFin' menu angle.
public	int	GetTotalElementsNumber	None	Calculate the total number of buttons in the category.
public	EtatMenu	GetActuelMenuState	None	Get the current menu's state.
private	float	ClampRotation	float rotation	Clamp a rotation between 0 and 360°.
public	Circular MenuCate gory	AddCategory	None	Create a new category, add it to the menu and return it.
public	void	RemoveCategory	CircularMenuCat egory _category	Remove specific category from the menu.

public	void	UpdateButtonsNumber	None	Update the number of buttons on the menu and calculate the radius of the menu if 'manuallySetRadius' is true.
public	Circular MenuButton	GetButtonFromName	string	Return a button with a similar name contains in this menu. Return null if there is no menu with this name.
public	Circular MenuCategory	GetCategoryFromName	string	Return a category with a similar name contains in this menu. Return null if there is no category with this name.