SimpleMan.Utilities Download

Standard utilities for any project on Unity engine.

Authors: Igor-Valerii Chebotar, Alexey Naumenko
Email: igor.valerii.chebotar@gmail.com

How to install plugin?

Open installer by the click on Tools -> Simple Man -> Main Installer -> [Plugins' name] -> Click 'Install' button. If you don't have one or more of the plugins this plugin depends on, you must install it first.

Execute once system

Gives ability to execute method only one time per frame, no matter how many calls was received.

Methods

Function name	Description
ExecuteOncePerFrame	Execute target method. Ignore other execution calls for this method in current frame.

C# Examples

```
//Will be called once
ExecuteOnceSystem.ExecuteOncePerFrame(DoAction)
ExecuteOnceSystem.ExecuteOncePerFrame(DoAction)
ExecuteOnceSystem.ExecuteOncePerFrame(DoAction)


//Works also with parameters
ExecuteOnceSystem.ExecuteOncePerFrame(( ) => DoAction("Hello"))
```

Component reference

Use this class to make a cached reference to component on current game object. This solution is better than standard C# fields and properties, because you don't need to call 'GetComponent' at awake or start

and check is component exist when you need to use it.

C# Examples

```
private ComponentRef<Health> _healthRef;


private void Awake()
{
    //Create instance of the reference at awake. Don't forget about this step
    _healthRef = new ComponentRef<Health>(gameObject);
}

public void ApplyDamage(float damage)
{
    //Thats way how you can get access to your 'Health' component.
    _healthRef.Value.Decrease(damage);

    //The component will be cached inside the reference and if it will be destroyed,
    //you receive message into console and exception in code. So you don't need
    //to check if component exist manually
}

//Try also ChildComponentRef and ParentComponentRef
```

Project and inspector window blocker

You can block the inspector and project windows using CTRL+SPACE and CTRL+SHIFT+SPACE hot keys

Collection extensions

Methods

Function name	Description
Random	Gives random element in collection
ForEach	Make action for each element in collection
Except	Returns collection without element
Validate	Returns collection without null elements
GetElementIndexByKey	Get index of dictionary pair with specified key

Function name	Description
AddUnique	Works with list, queue, stack and dictionary. Ignore *Add* action if collection already contains element
AssertNoNullElements	Throws an exeption when at least one element is null

C# Examples

```
//Add element only if it isn't exist in collection
targetsList.AddUnique(targetObject);


//Get random color from collecion
Color randomColor = colorsCollection.Random();


//Reset position for each game object in list
targetsList.Foreach(x => x.transform.position = Vector3.zero);
```

Transform extensions

Methods

Function name	Description
IsDirectChildOf	Retruns true if specified object is direct child of this object
IsDirectParentOf	Retruns true if specified object is direct parent of this object
GetDirectChildren	Get only direct children for this transform
GetDirectChildrenOfType	Returns array of direct children that have certain component
DestroyChildren	Destroy all children of current transform
DestroyChildrenImmediate	Destroy all children of current transform (for editor mode)
SetPositionAndRotation	Set position and rotation using *PositionAndRotation* structure
SetLocalPositionAndRotation	Set local position and rotation using *PositionAndRotation* structure
GetPositionAndRotation	Returns *PositionAndRotation* structure

Function name	Description
GetLocalPositionAndRotation	Returns *PositionAndRotation* structure

C# Examples

```
//Destroy all children
transform.DestroyChildren();


//Get direct children array
Transform[] children = transform.GetDirectChildren();


//Get only direct children with 'Health' component
Health[] children = transform.GetDirectChildrenOfType<Health>();
```

Component extensions

Methods

Function name	Description
TryGetComponentInChildren	Return true if at least one child of this object have certain component
TryGetComponentInParent	Return true if at least one parent of this object have certain component

C# Examples

```
//Throws exception if parent object don't have 'Animator' component
if(TryGetComponentInParent<Animator>(out Animator animator) == false)
    throw new NullReferenceException("Animator was not found in chilren");
```

Object extensions

Methods

Function name	Description
FindObjectOfType	Returns first object of specified type on scene (interfaces supported)
FindObjectsOfType	Returns array of objects of specified type on scene (interfaces supported)
With	Pseudo-builder
PrintLog	Print debug log message with name of the object caller
PrintWarning	Print debug log warning message with name of the object caller
PrintError	Print debug log error message with name of the object caller
SetSquarePrefix	ObjectName -> [Prefix]ObjectName
GetNameWithoutSquarePrefix	Returns name of the target game object without prefix
SetUnderscorePrefix	ObjectName -> Prefix_ObjectName
GetNameWithoutUnderscorePrefix	Returns name of the target game object without prefix
ToScene	Move object to the target scene

C# Examples

```
//Throw exception
if(_health == null)
    this.PrintWarning("Component 'Health' not exist");


//Game object name will look like this '[Player]PreviousName'
this.SetPrefix("Player");


//Find interactable object on scene
IInteractable interactable = this.FindObjectOfType<IInteractable>();
```

Base types extensions

Methods (object)

Function name	Description
(object) Exist	Returns true if object is not null
(object) NotExist	Returns true if object is null

C# Examples

```
//Before
if(target != null)
{
    ...
}

//After
//Supports also UnityEngine.Object and their special
//null checking
if(target.Exist())
{
    ...
}
```

Methods (string)

Function name	Description
FirstCharToUpper	sad but true -> Sad but true
ToSplitPascalCase	SadButTrue -> Sad But True
WithUnderscorePrefix	ObjectName -> Prefix_ObjectName
WithoutUnderscorePrefix	Prefix_ObjectName -> ObjectName
WithSquarePrefix	ObjectName -> [Prefix]ObjectName
WithoutSquarePrefix	[Prefix]ObjectName -> ObjectName
ExctractFileNameFromPath	Assets/Data/File.Asset -> File
RemoveClone	SomeObject (Clone) -> SomeObject

Methods (int and float)

Function name	Description
(float, int) ClampPositive	Return closest positive value
(float, int) Clamp01	Return value from 0 to 1
(float, int) Abs	Returns absolute value
(float, int) InRange	Returns true if value within target range
(float, int) OutOfRange	Returns true if value out of target range
(float) ClampAsAxis	Return value from -1 to 1
(float) Round	Returns closest value to int
(float) RoundToInt	Returns closest value to int and converts it to int
(float) Floor	Returns the largest integral value less than or equal to the specified number
(float) FloorToInt	Returns the largest integral value less than or equal to the specified number and converts it to int
(float) Ceil	Returns the smallest integral value greater than or equal to the specified number
(float) CeilToInt	Returns the smallest integral value greater than or equal to the specified number and converts it to int

C# Examples

```
private float _health;

//You will never see Health value as negative,
//because method 'ClampPositive' clamps value
//from 0 to positive infinity
public float Health
{
    get => _health;
    set => _health => value.ClampPositive();
}


private List<GameObject> _targets;
```

```
public void GetTargetAtIndex(int index)
{
    //Throw exception if index is out of range.
    //Before
    if(index < 0 || index > _targets.Count)
        throw new OutOfRangeException("Index is out of range");

    //After
    if(index.OutOfRange(0, _targets.Count)
        throw new OutOfRangeException("Index is out of range");

    return _targets[index];
}
```

Methods (Vector2 and Vector3)

Function name	Description
(Vector2) XY2XZ	Return Vector3 as projection of XY plane to XZ
(Vector3) XZ2XY	Return Vector2 as projection of XZ plane to XY. Y value will be ignored

C# Examples

```
//Get projection of position on XY plane
Vector2 position2D = transform.position.XZ2XY();
```

Methods (Color)

Function name	Description
Invert	Return inverted color
MaxAlpha	Return the same color with maximum alpha
MinAlpha	Return the same color with zero alpha

Methods (Matrix4x4)

Function name	Description
ExtractRotation	Return Quaternion rotation value from transform matrix
ExtractPosition	Return Vector3 position value from transform matrix

Function name	Description
ExtractScale	Return Vector3 scale value from transform matrix

Mathematics

Methods

Function name	Description
GetClosest	Return closest Transform or other component to target point
GetPointsOnCircle	Return array with points positions
GetPointOnCircle	Return point position on circle by angle

C# Examples

```
//Get items around
IInteractable[] items = GetAvailableItems();

//Get closest interactable item to player
Vector3 playerPosition = transform.position;
IInteractable closestItem = Mathematics.GetClosest(playerPosition, items);



//Get angle between horizontal axes and mouse input
float inputAngle = Vector2.Angle(inputAxes, Vector2.right)

//Set top down crosshair position
_crosshair.transform.position =
    Mathematics.GetPointOnCircle(
        transform.position,
        _crosshairDistance,
        inputAngle).
        XY2XZ();
}
```