uPano

Version 2.2

Infinity Code, 2018-2020

https://infinity-code.com/

Overview

Introduction

uPano (Unity Panoramic Framework) is a universal solution for displaying dynamic and static panoramas, and creating virtual tours.

Most of the existing types of panoramas are supported: spherical, cylindrical, cubic panoramas on single or six images, cubemap.

uPano is very easy to learn and use, and is great for people who do not have programming experience. In most usage scenarios, you can make interactive panoramas without creating your own scripts (for more details, see Interactive Element / Events section).

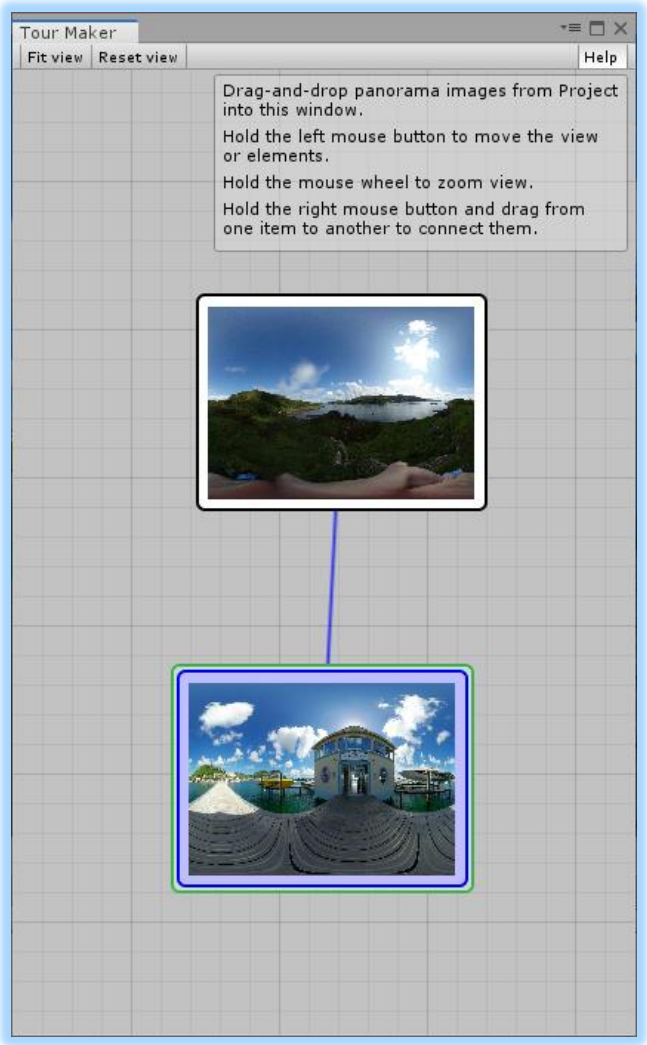
If you have some very specific purpose, uPano has a powerful and easy-to-use API that will allow you to implement any behavior.

Supports Unity v2017.4 LTS or later.

Supported platforms: Standalone, iOS, Android, Universal Windows Platform, WebGL.

Other platforms have not been tested, but most likely uPano will work well.

Quick tour creation



Select Window / Infinity Code / uPano / Tour Maker.

Click «Create a new tour». This will create a new empty tour with default settings.

If you are making a tour for Google VR, select Hierarchy / uPano Tour / Tour (Script) / Preset - Google VR. This will remove the keyboard and mouse controls, add Timed Gaze plugin, and apply the required settings for VR.

Drag and drop panorama textures from Project to Tour Maker.

A separate panorama will be created for each texture.

Right-click on the panorama with which you want to start the tour, and select Set Start Panorama in the context menu. The start panorama will be highlighted in green.

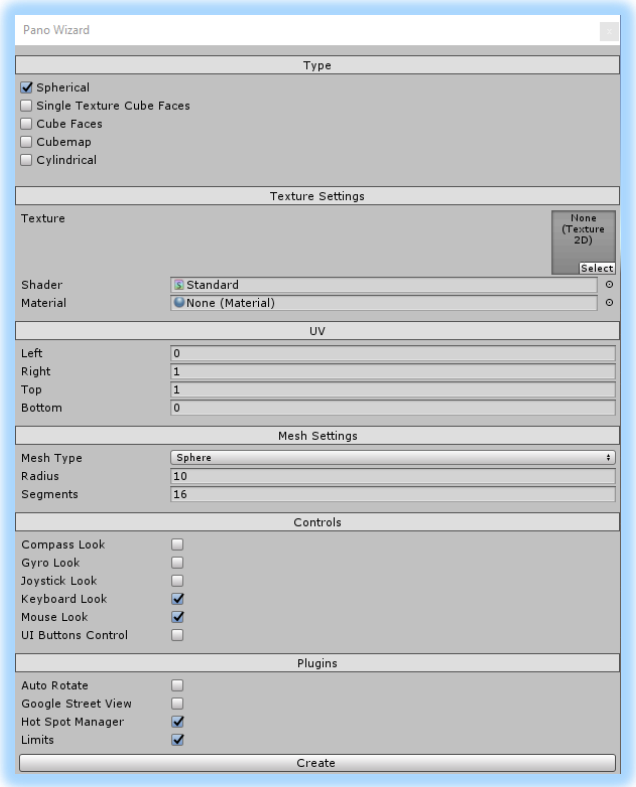
To create an interactive element for switching between panoramas, right-click on the panorama from which you want to switch, drag to the target panorama, and release the right mouse button.

This will open Visual Element Editor in connection mode.

Right-click in the place where you want to create an interactive element, and select the element type (HotSpot or Direction).

In Tour Maker, outgoing transitions from the selected panorama will be highlighted in blue.

Quickly create a panorama (wizard)



Select

Hierarchy / Create / 3D Object / Panorama

or

Window / Infinity Code / uPano / Wizard

to open the panorama wizard.

Select the type of panorama, specify the texture, enable the controls and plugins you need, and click «Create».

uPano will create a fully-ready panorama.

Just start the scene, and enjoy the result.

Structure

uPano consists of:

Pano - main panorama script. Contains the panorama orientation (pan, tilt) and camera settings. Must be present in every panorama.

Pano Renderers - scripts to displaying the panorama type. Contains settings for mesh, texture, material, shader and UV. Each panorama should use one Pano Renderers.

Controls - scripts that implement control of the panorama using input sources.

Plugins - scripts that implement additional features.

Using controls and plugins is optional.

You can use them in any combination, or not use at all.

Viewing parameters

To position any point on the panorama, two values are used:

Pan - the horizontal angle of rotation of the camera is 0-360 degrees clockwise, where 0 is north, 90 is east, 180 is south, 270 is west.

Tilt - deviation angle «up» and «down» from the horizon.

The range is from -90 to 90 degrees, where 0 is horizontal, 90 is up, and -90 is down.

The point on the panorama that the camera is looking at is called Point of View (POV).

Important: changing the tilt in a cylindrical panorama moves the camera up / down instead of rotating.

The camera has two additional parameters:

Fov - the vertical field of view of the camera in degrees.

North Pan - the horizontal angle of rotation of the camera is 0-360 degrees clockwise to the true north.

Cameras

uPano requires a camera to properly display and interact with the panorama.

To interact with interactive elements, the camera must have a Physics Raycaster component.

By default, uPano creates a new camera to display the panorama.

But if you need, you can specify an existing camera.

For example, this is very useful when creating a VR application.

To work with multiple cameras, use Multi Camera component.

To work with cameras that have an orthographic projection, use Orthograpthic Cameras component.

Orientation of a panorama

uPano supports panorama orientation in three modes:

1. **Rotate Camera** (by default) - when changing pan and tilt, the camera that displays the panorama will be rotated. In most cases, it is recommended to use this mode.

2. **Rotate Panorama** - when changing pan and tilt, the GameObject of the panorama will be rotated.

3. **Rotate GameObject** - when changing pan and tilt, the specified GameObject will be rotated. This is useful for VR applications when you want to rotate multiple cameras simultaneously.

Important: For the camera created by uPano, only Rotate Camera mode is available. If you want to change the orientation mode, specify an existing camera.

UV

For all types of panoramas, except for a cubic panorama on six images and a cubemap, you can adjust UV of the panorama.

For a cubic panorama on single image, you can rotate the side texture to 90, 180 or 270 degrees.

Proper adjustment of UV is very important for a cubic panorama on single image. Even a small error can give a visible seam on the faces of the cube, so be careful when setting up UV.

In most cases, UV for the other types of panoramas do not need to be adjusted. But still it can be useful for solving specific problems. For example, to clip an empty area at the bottom of an image downloaded from Google Street View by third-party ways.

Texture Compression

When you use a texture that you have in a project, the compression and format of the textures depends on your texture import settings.

When you download a texture using Downloader, Google Street View, or set Texture2D using uPano API, the texture compression depends on Pano Renderer / Compress Texture field.

If texture compression is enabled, the texture will be automatically compressed to DXT1 or DXT5 format, which greatly reduces memory usage.

Scriptable Render Pipeline (LWRP, HDRP)

uPano automatically detects the use of SRP, and if necessary, will ask you to enter the shader parameter where the texture should be set.

WebGL

When you'll try to use Google Street View, you may encounter a CORS (Cross-Origin Resource Sharing) restriction.

https://en.wikipedia.org/wiki/Cross-origin_resource_sharing

Currently there is only one way to work around this - use PHP bridge (proxy).

Example and instructions you can download here:

https://infinity-code.com/downloads/uPanoCORS.zip

Panorama types (Pano Renderers)

Spherical

The basis of the spherical panorama is an image collected from a variety of individual frames in a spherical (equidistant) projection. A characteristic feature of spherical panoramas is the maximum possible viewing angle (360 × 180 degrees).

uPano can display this type of panorama on a sphere or icosahedron.

Icosahedron is a regular convex polyhedron, each face of which is an equilateral triangle. Using icosahedron allows you to improve the display of the panorama at the top and bottom points.

To display spherical panoramas, use Spherical Pano Renderer.

Cylindrical

Cylindrical panorama is characterized by a viewing angle of 360 degrees horizontally and a viewing angle of less than 180 degrees vertically. In other words, a cylindrical panorama is a variant of a spherical panorama, devoid of zenith and nadir (images of the highest and lowest points of view).

To display cylindrical panoramas, use Cylindrical Pano Renderer.

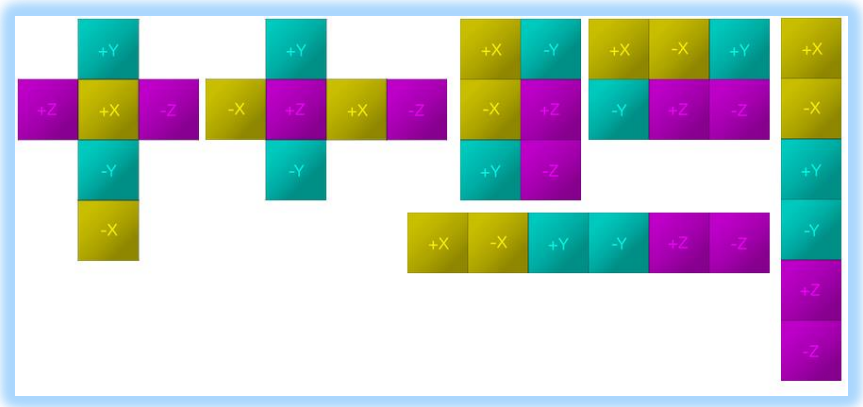
Cubic panorama on six images

Cubic panorama is obtained by projecting images onto the inner faces of the cube. You can imagine an observer inside a huge cube, on the inner walls of which are glued images of the appropriate scale. Cubic panorama allows you to create panoramas with viewing angles of 360 degrees horizontally and 180 degrees vertically. Like a spherical panorama, the cubic panorama is prone to distortions.

To display cubic panoramas on six images. use Cube Faces Pano Renderer.

If you see the seams on the edges of the cube, select textures in Project window and set Wrap Mode - Clamp.

Cubic panorama on single image



In this type of panorama, the images of the sides of the cube are combined into one large image.

There are three main types of layouts:

1. Crosses

2. Lines

3. 3 rows, 2 columns

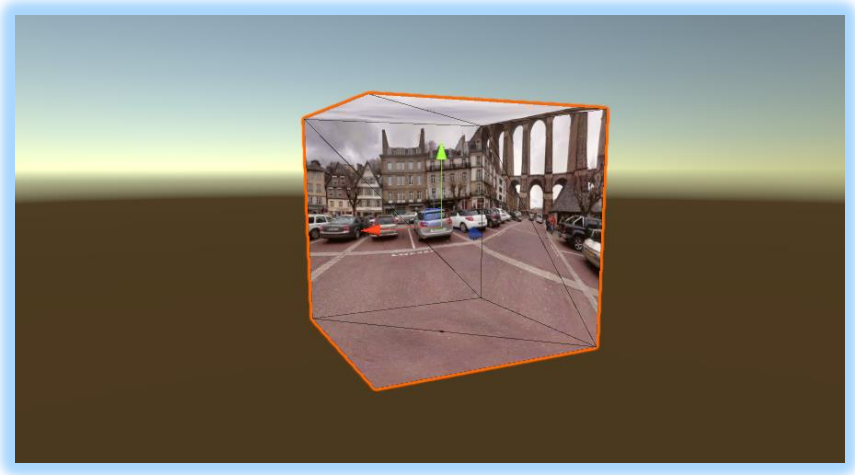
Each type has a horizontal and vertical orientation.

Sides of the panorama can be reflected and / or rotated by a fixed angle (90, 180, 270).

uPano has presets for the main types of layouts and allows you to adjust UV of each side of the panorama.

To display cubic panoramas on single image, use Single Texture Cube Faces Pano Renderer.

Cubemap



uPano can display an existing cubemap as a panorama.

https://docs.unity3d.com/Manual/class-Cubemap.html

To display cubemap panoramas, use Cubemap Pano Renderer.

Controls

Components that allow the user to control the panorama.

You can use an unlimited number of different controls in any combination.

Mouse Control

Implements panorama control using the mouse and touch (on mobile devices).

Modes:

1. Drag (by default) - the panorama will be moved by dragging. Supports inertia when you stop dragging.
2. Free - the panorama will constantly move towards the cursor.
3. Left Mouse Button Down - the panorama will move towards the cursor while you hold the left mouse button.

Implements the changing of fov using the mouse wheel and pinch to zoom (on mobile devices).

Keyboard Control

Implements panorama control using the keyboard.

To change pan and tilt, use the horizontal and vertical input axes.

https://docs.unity3d.com/Manual/class-InputManager.html

To change fov, use «+» and «-» buttons on the main or numeric keypad.

You can adjust the sensitivity of changing the values and limit the rotation axes of the panorama.

Tip: To prevent pan, tilt or fov from changing using the keyboard, set the sensitivity to 0.

Compass Control

Implements panorama control using a device compass.

This control changes only pan.

Gyro Control

Implements panorama control using a device gyro.

This control changes pan and tilt.

UI Buttons Control

Implements panorama control using the on-screen buttons.

Automatically integrates with Auto Rotate plugin.

Joystick Control

Implements control panorama uing the on-screen joystick.

Drag the center part of the joystick and the panorama will move in the same direction.

UI Compass Control

Compass showing the current orientation of the panorama.

When you click on the center, it sets the orientation to the north.

When you click on the side arrows, the panorama rotates 90 degrees clockwise or counterclockwise.

Plugins

Scripts that implement additional features of panoramas.

Formally, all control, managers of interactive elements and some other components are also plug-ins, but they have a lot of features and will be described in separate sections.

Auto Rotate

Automatically rotates the panorama clockwise.

When the user interacts with the panorama using any control, the automatic rotation stops.

After 10 seconds of inactivity, automatic rotation resumes.

If tilt is not equal to zero, then this value will gradually tend to zero.

Automatically integrates with UI Buttons Control.

Downloader

Downloads the panorama texture at the specified URL.

Optionally can download low resolution texture first.

> **Important:** A low resolution texture must be of the same type as a high resolution texture.

Supports all Pano Renderers, except Cubemap.

Limits

Allows you to limit the valid values of pan, tilt, and fov.

Strongly recommended for use.

Multi Camera

Synchronous changes fov for multiple cameras.

Additionally, it can change the local position of cameras depending on fov.

This is useful for VR applications to prevent the problem of too much divergence of images for the left and right eyes when fov value is small.

Orthographic Cameras

Extends Multi Camera plug-in, allowing cameras to have an orthographic projection.

Additionally, it has the ability to customize the conversion of fov - orthographicSize values.

Timed Gaze

Automatically clicks on interactive elements (HotSpots, Directions) after the specified hover time.

Interactive Elements

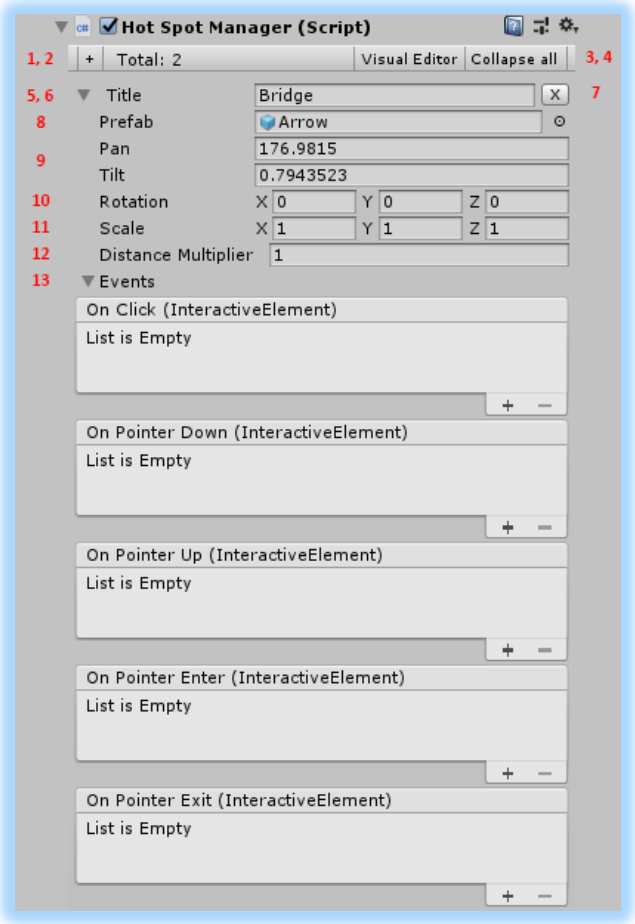
Interactive element is an object in the panorama, with which the user can interact.

Interactive elements support events and actions, and can be programmed for any behavior.

Hot Spot

Hot Spot Manager component adds hot spot support to the panorama.

Hot spot is an interactive GameObject, which is positioned on the panorama using pan and tilt.



Hot Spot Manager contains:

1. **Button «+»** - add a new hot spot.
2. **Total: X** - the number of elements.
3. **Visual Editor** - opens a visual editor.
4. **Collapse all** - Collapses all items and their events.
5. **Arrow down** - collapse item fields.
6. **Title**. This name will set to the GameObject - a hot spot instance.
7. **Button «x»** - remove the hot spot.
8. **Prefab** - GameObject, a copy of which will be used for hot spot.
9. **Pan, tilt** - position hot spot on the panorama.
10. **Rotation** - local rotation of the hot spot instance.
11. **Scale** - local scale of the hot spot instance.
12. **Distance Multiplier** - the multiplier of the distance from the center of the panorama to the point on the mesh. Allows you to move the hot spot closer if it goes beyond the panorama.
13. **Events** - hot spot events.

Direction

Direction Manager component adds support for direction to the panorama.

Direction is an arrow that points in the direction of another panorama. This is positioned in the panorama using the pan.

In general, Direction Manager has the same fields as Hot Spot Manager, except for the settings for all directions:

Default Prefab - GameObject, an instance of which will be used for direction, if no prefab for the element is specified.

Internal Radius - the radius from the central point of rotation to the instance of the direction.

External Radius - the radius from the camera to the center point around which direction will be located.

Vertical Offset - vertical offset of the center point.

Visual Editor

Visual editor makes it easy to manage interactive elements on the panorama.

Important: In the current version, the visual editor supports only spherical, cylindrical and cubic panoramas on single image.

Click «Window / Infinity Code / uPano / Visual Interactive Element Editor» or «Visual Editor» in the interactive element manager inspector to open the visual editor.

Select in the «Hierarchy» or «Project» window any GameObject containing a panorama to display the used texture and interactive elements.

You can drag and scale the viewing area.

Click «Reset view» to reset the movement and scale of the viewing area.

To create an interactive elements, right-click and select «Create {TYPE OF ELEMENT}» from the context menu. A new interactive element will be created under the cursor.

You can drag element to any available panorama area. But you cannot drag an interactive element outside UV of the panoramas.

To remove element, right click on it and select «Remove» from the context menu.

Click on the interactive element using the left mouse button to open the interactive element settings window.

This window displays the settings only for one element, so it is much more convenient to use it in Interactive Element Manager inspector.

Above the interactive element icon displayed title.

For hot spot: If the title is red, then prefab is not specified and hot spot will not be displayed.

Events

Interactive elements have five events:

Clicking, pressing the left mouse button (or touch), releasing the left mouse button (or touch), the cursor is enter to the interactive element, the cursor is out from the interactive element.

For each event, you can add an unlimited number of actions.

Action is a public method of a class that inherits MonoBehaviour, which takes an interactive element as a parameter.

You can create your own actions, or use a predefined actions.

Global events

If your interactive elements have the same actions for an event, you can use global events that will be called for all interactive elements of the same type before the element's events.

To use global events, add the global event component of the appropriate type.

For hot spot: HotSpotGlobalActions.

For direction: DirectionGlobalActions.

If some interactive element should ignore global events, you can specify it in the settings of the element.

Predefined actions

Predefined actions are: used for all types of interactive elements and used only for hot spots.

For all types of interactive elements:

Copy Pan Tilt

Destroy Current Panorama

Instantiate Prefab

Load Another Panorama

Load Scene

Open URL

Play Sound

Set Cursor

Set Fov

Set GameObject Active

Set Pan and Tilt

Set Scale

Set Text

Set Transform Position

Set Transform Rotation

Only for hot spots:

Destroy Current Hot Spot

Set Hot Spot Color

Set Hot Spot Rotation

Show TextMesh Tooltip

Show Tooltip

Show Tooltip Multi Camera

All predefined actions have a public Invoke method, which must be called to perform the action.

> **Note:** Show Tooltip Show Tooltip Multi Camera and Show TextMesh Tooltip actions have three public methods: Invoke and Show - show tooltip, Hide - hide tooltip.

In the future, the list of predefined actions will expand.

If you have suggestions, what other actions need to be added, please contact us.

Quick actions

For interactive elements, there are the same actions that you will most likely use for most elements.

You can create quick actions in Quick Actions section of the settings of the interactive element.

Load another panorama on click:

Specify Load Panorama Prefab and Load Another Panorama action will be automatically added to the interactive element.

Switch to another panorama in the scene on click:

Specify Switch To Panorama and Set GameObject Active action will be automatically added to the interactive element.

Showing a tooltip:

Enter Tooltip text, select the action, and specify Tooltip Prefab.

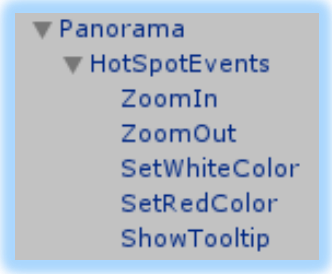
For the interactive element, the specified action will be automatically added.

Interactive elements without programming

uPano allows you to specify the behavior of interactive elements without programming.

Step by step instructions on the example of hot spot:

Create a child GameObject for the panorama, which will contain hot spot events.

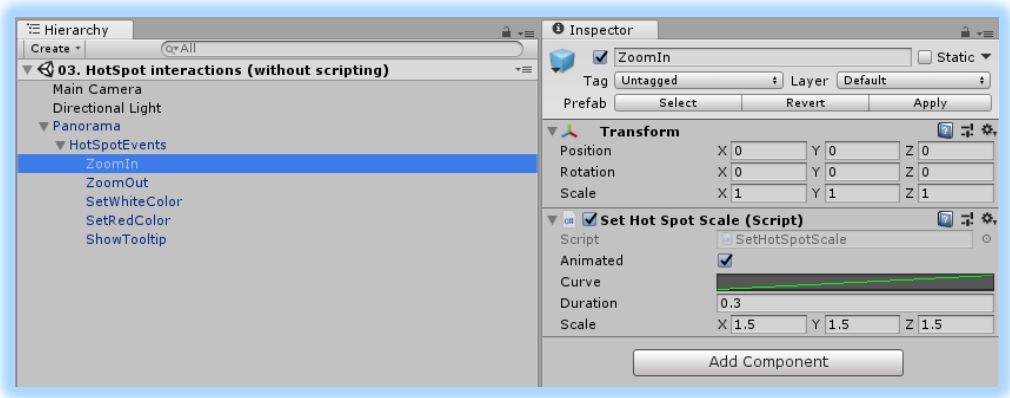


Tip: Create separate GameObject for each action. This will help you more easily navigate in action.

Important: If you want to make a prefab of the panorama, the action must be on the child of the panorama GameObject. Prefab cannot reference objects in the scene outside the prefab.

Select GameObject that will store the action.

Add one of the predefined actions, or your own script containing the action.



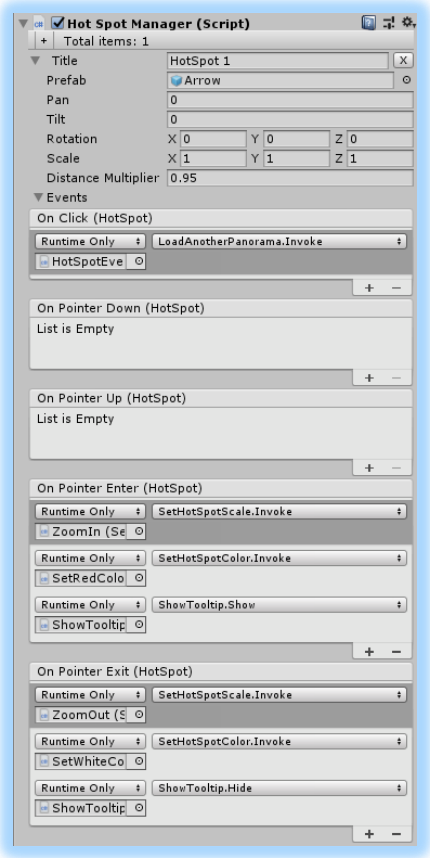
In Hot Spot Manager, create a new Hot Spot.

In the event that should perform this action, press «+».

Specify GameObject that contains the action.

Select the action script and Invoke method if you use predefined actions, or your own method that should be called.

Note: the method must contain the interactive element parameter, in which the element that triggered the action will be passed.



Transitions

uPano has a built-in system of compound transitions, allowing you to easily create complex unique transitions without programming.

For each action that supports transitions (DestroyCurrentPanorama, LoadAnotherPanorama, LoadScene), you can specify two transitions:

Before - will be invoked before the action.

After - will be invoked after the action.

> **Note:** you need to specify not the transition itself, but Prefab which contains the transition.

All available transitions are in «Component / uPano / Transitions».

Simple transitions

All simple transitions are animated and smoothly change the state from initial to final.

Available simple transitions:

Changing panorama values: Set Fov, Set Pan and Tilt.

Changing interactive elements: Look At Active Element, Scale Interactive Elements.

Full-screen overlays: Blur, Tint.

> **Important:** Look At Active Element and Scale Interactive Elements (with Element Type except All Elements) cannot be used in After Transition, because the element that causes the transition does not exist anymore.

Compound transitions

Compound transitions are realized by combining several simple transitions using helper and combined transitions.

Compound transitions are located in «Component / uPano / Transitions / Helpers».

Consecutive Transitions - an array of transitions, each element of which begins after the end of the previous one. The transition ends when the last element is completed.

Simultaneously Transitions - an array of transitions, all elements of which start at the same time. The transition ends when the longest element is completed.

Delay - helper transition, to implement the delay in other combined transitions.

Step-by-step example of how to create a compound transition

In this example, we will create two complex transitions.

The first transition will simultaneously center and zoom the interactive element that the user clicked on, and blur the panorama.

The second will perform the opposite action.

Before Transition:

Create a new GameObject (Hierarchy / Create Empty) and rename it to BeforeTransition.

Create a child GameObject (Hierarchy / Create Empty Child) and rename it to Phase 1.

Add Look At Active Element transition, enable Use Fov, and set Target Fov - 10.

Select BeforeTransition and create a child with name Phase 2.

Add Blur transition and set Blur Material - TransitionBlurMaterial.

Select BeforeTransition and add Simultaneously Transitions.

Set Transitions / Size - 2, and select Element 0 - Phase 1, Element 1 - Phase 2.

In Project, create Transitions folder, and then drag BeforeTransition from the Hierarchy to this folder.

BeforeTransition name in Hierarchy should be blue. This means that prefab is created.

Remove BeforeTransition from Hierarchy.

After Transition:

Create a new GameObject and rename it to AfterTransition.

Create a child GameObject and rename it to Phase 1.

We cannot use Look At Active Element, because the element that caused the transition no longer exists.

Add Set Fov transition, and set From Fov - 10, To Is Original - ON.

Select AfterTransition and create a child with name Phase 2.

Add Blur transition, set specify Blur Material - TransitionBlurMaterial, From Radius - 30, To Radius - 0.

Select AfterTransition and add Simultaneously Transitions.

Specify Transitions / Size - 2, and select Element 0 - Phase 1, Element 1 - Phase 2.

Drag AfterTransition to Project / Transitions.

Remove AfterTransition from Hierarchy.

Using transitions:

Create a new hot spot and expand Quick Actions.

In Target Panorama select the prefab of the panorama to which you want to go.

Two new fields will shown: Before Transition Prefab and After Transition Prefab.

Select the appropriate prefab from Project / Transitions.

That's all, it remains to verify that this works.

Start the scene and click on the hot spot to see the transitions.

Additional features

Video panoramas

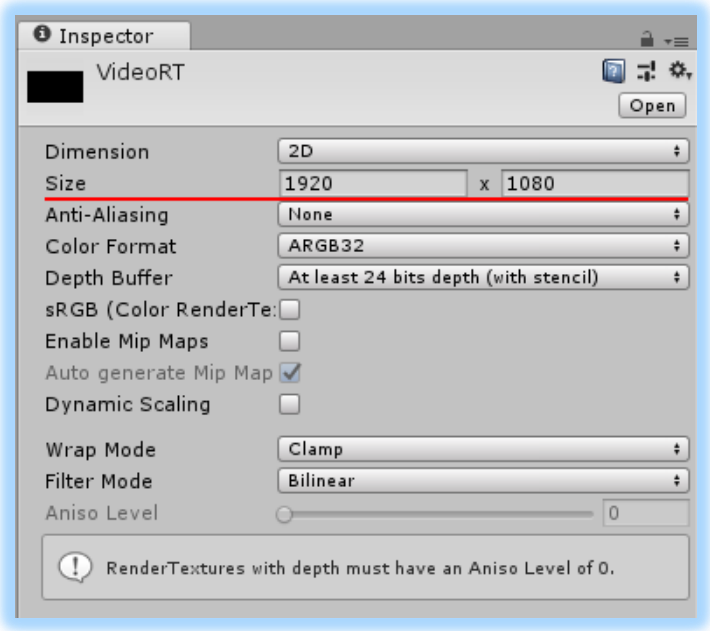
uPano can display video panoramas.

To use a video panorama:

Import the video into the project.

Create a new Render Texture (Assets / Create / Render Texture).

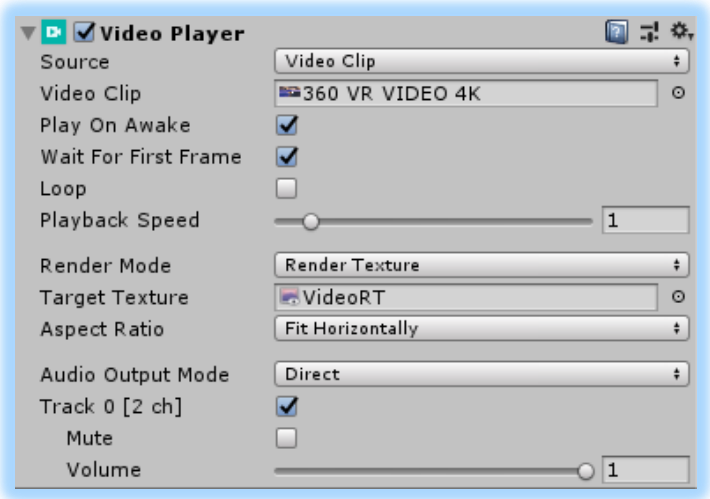
Specify the size of the Render Texture to be equal to the video resolution.



In the scene, create a new GameObject, and add Video Player component to it.

In Video Clip field, specify a video.

Select Render Mode - Render Texture and specify the early created Render Texture in Target Texture field.



In Pano Renderer, set Render Texture to Texture field, and adjust UV.

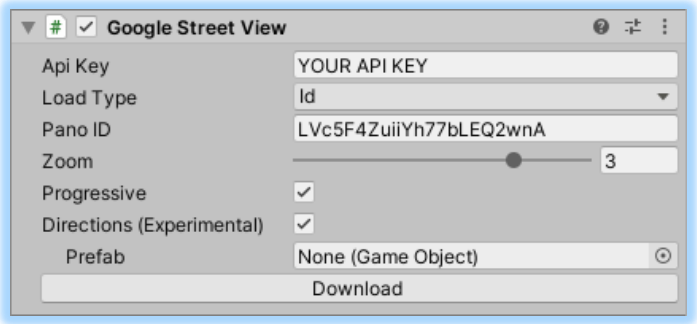
When you start the scene, you will see a video panorama.

Tip: For Youtube 360 video, you need to use Single Texture Cube Faces Pano Renderer.

UV Preset - Youtube (3x2).

Google Street View

uPano can download panorama from Google Street View by panorama ID, or by location (latitude and longitude in decimal).



Important: Google Street View requires Spherical Pano Renderer.

The component has two work scenarios:

1. **Dynamic.** When you start the scene, the image from Google Street View will be downloaded and used as a texture of the panorama.

2. **Static**. Click «Download» to download the panorama and save it in the project.
After the panorama image is downloaded, this component should be removed from GameObject.

Size of a panorama texture:

Width = 512 * 2 ^ zoom.

The height of the panorama is not constant, and is calculated based on the colors of the loaded image.

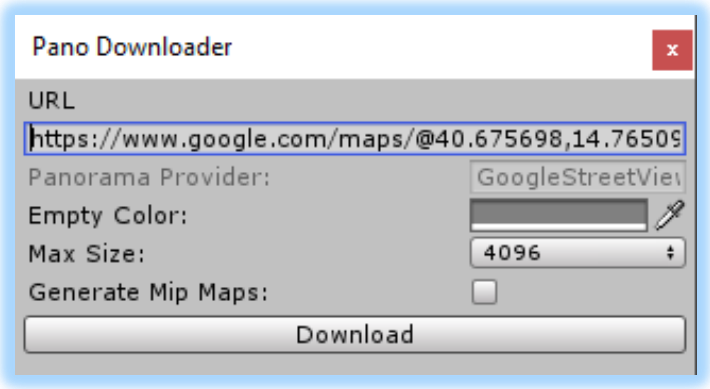
When using progressive download, uPano will first download the panorama in low resolution (zoom - 0), then it will download the panorama in higher resolutions (zoom - 1, 2, 3) up to the specified zoom value.

When using Directions - ON, uPano will display directions to neighboring panoramas.

Important: this experimental feature, and in some cases may work unstably.

Pano Downloader

A tool for downloading panoramas from Google Maps and Yandex Maps by URL.



Open the browser and go to Google Maps or Yandex Maps site.

Use the map to open the panorama.

Copy URL from your browser's address bar.

Open Window / Infinity Code / uPano / Pano Downloader.

Paste URL into URL field and click Download.

The downloaded panorama will be saved in the folder «Assets / Panoramas».

Integration with third-party assets

Third-party assets are not included in uPano package and you must buy them separately.

For integration, you need to have uPano and another asset in the project.

Online Maps

https://assetstore.unity.com/packages/tools/integration/online-maps-v3-138509

Integration with Online Maps allows you to display Google Street View panoramas using uPano.

To enable integration, add Online Maps Pano Connector to the map GameObject and click Enable uPano.

The map will show overlay for which Google Street View panoramas are available.

When you click on the map, uPano will download and display the panorama of the location.

Playmaker

https://assetstore.unity.com/packages/tools/visual-scripting/playmaker-368

Integration with Playmaker allows you to use the uPano API in visual mode.

To enable integration import «uPano-Playmaker-Integration-Kit.unitypackage» into the project.

You can do this in two ways:

1. Window / Infinity Code / uPano / Playmaker Integration Kit
2. Manually import the package located «Assets / Plugins / Infinity Code / uPano / Packages / uPano-Playmaker-Integration-Kit.unitypackage».

Playmaker will add new actions for uPano, grouped into sections starting with uPano.

uPano API

For more information about all public classes, methods, and variables, see API Reference.

Online version:

https://infinity-code.com/docs/api/upano/

The local version of API Reference is located in «Plugins / Infinity Code / uPano / Documentation / API Reference (Do not unpack in assets folder).zip».

Important: do not unpack the archive into Assets folder of your project.

The order of the values in the point on the panorama:

Vector2, Vector3: X - pan, Y - tilt;

Parameters of methods: first - pan, then - tilt.

The order of values in geographic coordinates:

Vector2, Vector3: X - longitude, Y - latitude;

Parameters of methods: first - longitude, then - latitude.

Making changes to the source code uPano

We do not recommend that you make changes to uPano code yourself, because you will lose it on update.

Most tasks can be solved using API.

If you really need to modify the source code of uPano, please send us your changes. We will include your changes into the project, or we will give you the best way to do it.

Updating

We send only stable versions to Unity Asset Store.

The update period is several months.

uPano has a built-in update system, using which you can download the latest versions.

Important: Always make backups of the project before updating assets.

Select

Window / Infinity Code / uPano / Check Updates

or

uPano / Help / Check Updates

to open the built-in update system.

Specify your Invoice Number or Order Number.

Select the update channel and click «Check New Versions».

You can find out your Invoice Number in Unity Asset Store order confirmation email, or page:

https://assetstore.unity.com/orders

Each channel shows a maximum of 10 latest updates.

If updates are available, then you can read the list of changes and download it.

If you have problems installing the update, then:

1. Open an empty scene.

2. Delete folder «Standard Assets / Infinity Code / uPano».

3. Import the new version of uPano into the project.

If you want to return to the previous version of uPano, then select the «Stable Previous» channel. Using this channel you will be able to get 10 previous stable versions.

uPano automatically checks for updates every 24 hours.

If the new version is available, you will see a red label in uPano Inspector header. Clicking on it will open the built-in update system.

Automatic check of the update does not require Invoice Number and uses the last selected update channel. If the update channel is not selected, only stable versions will be checked.

Support

We provide support by email (support@infinity-code.com) in English and Russian, or on the forum (https://forum.infinity-code.com) in English.

If something does not work for you, you found a bug, or you have any suggestions, please contact to us.

Please don't forget to specify your version of Unity, OS and the current version of uPano.

The usual response time is from a few minutes to 24 hours.

We manually control the spam folder, so do not worry, we will not miss your request.

If you did not receive a response within 24 hours:

1. Check the spam folder.

2. Contact us using another way.

Other Infinity Code assets

Huge Texture

https://assetstore.unity.com/packages/tools/input-management/huge-texture-163576

Huge Texture allows you to import and use textures larger than 8192x8192px.

How it works:

When importing a texture, Huge Texture splits the texture into pages and saves it as a Texture Array. Texture Array is combined on the shader side, which does not create extra draw calls and has almost no effect on performance.

Features:

- The maximum size of the texture: PNG and JPEG up to 16384x16384px, RAW up to 2GB (0.715 gigapixels, 26624x26624px for square textures without transparency);
- Standard Render Pipeline, Universal Render Pipeline (URP, LWRP), High Definition Render Pipeline (HDRP);
- Compressed and uncompressed formats;
- Does not produce extra draw calls. You will have as many draw calls as you would with a regular texture;
- API for working with huge textures, as with regular Texture2D;
- Built-in update system.

Requirements:

- Not all platforms support Texture Arrays.
  The list of supported platforms is here:
  https://docs.unity3d.com/Manual/SL-TextureArrays.html
- The field must accept Texture (not Texture2D), and the component must accept custom material or shader.
  For example, Huge Texture can be used in Mesh Renderer, Raw Image, etc.

Mesh to Terrain

https://assetstore.unity.com/packages/tools/terrain/mesh-to-terrain-7271

Mesh to Terrain is a tool for easily and quickly converting a 3D terrain model created in 3ds Max, Terragen or any other editor to Unity Terrains.

Mesh to Terrain can convert textures to SplatPrototypes (Terrain Layers), generate terrain from several models and split the model into several terrains.

Features:

- Supports Unity v5.6 and higher;
- Unlimited number of models and terrains used in the component;
- Terrains are created in the same place where the meshes;
- Convert the textures of models to textures of terrain;
- Seamless result, when working with multiple terrains;
- Manually adding models or automatic detection of all models in the layer;
- No need to manually add the components of physics;
- Works with Relief Terrain Pack (optional);
- Built-in update system.

Online Maps

https://assetstore.unity.com/packages/tools/integration/online-maps-v3-138509

Online Maps is a universal multi-platform mapping solution for your 2D, 3D, AR / VR and mobile applications and games.

Fully customizable, incredibly easy to learn and use, and at the same time is one of the most powerful and flexible solutions in the industry.

Supports a huge number of services for any mapping needs, and has integration with the best Asset Store assets.

The package contains the complete source code without dependencies, and if you want to add or change some feature, you can easily do it.

Don't have programming experience or don't know C# - Online Maps supports visual scripting using Bolt and Playmaker.

All the features to create any map in Unity in one asset.

Features:

- Unity 2017.4 LTS and higher;
- Standalone, Android, iOS, Windows Store, WebGL;
- Online and offline maps;
- 2D maps and 3D maps with elevation;
- 2D, 3D, billboard and custom markers;
- You can display the map anywhere: on UI, in a scene, or draw into texture;
- Huge number of predefined tile sources: Google Maps, Mapbox, ArcGIS, Nokia Maps, Bing Maps, Open Street Maps, and many others (16 providers, 88 map types);
- Ability to create your own map style or use your own source of tiles (e.g. WMS);
- Multilingual map with or without labels;

- **Support Google API web services**: Direction API, Elevation API, Geocode API, Places API, Places Autocomplete API, Roads API;
- Other web services: AMap Search, Bing Maps Elevation API, Bing Maps Location API, HERE Routing API, Open Route Service Directions, Open Route Service Geocoding, Open Street Map Nominatim, Open Street Map Overpass API, QQ Search, What 3 Words;
- Show Google Street View using uPano;
- Additional Features: GPS (with emulator), cache, traffic, Overlays, Drawing API, runtime 3D buildings;
- Integration with: Bolt, Curved UI, EasyTouch, Fingers - Touch Gestures, NGUI, Playmaker, Real World Terrain, TouchScript, uContext, uPano;
- Easy-to-use and powerful API. Large number of examples of using is attached. Atlas of Examples;
- Built-in update system.

Real World Terrain

https://assetstore.unity.com/packages/tools/terrain/real-world-terrain-8752

Real World Terrain is a tool for automatically creating high-quality terrains, meshes, Gaia stamps and RAW files based on real-world data with global coverage.

Incredibly fast and easy to use, and allows you to create high-quality terrains in a couple of clicks.

In addition, Real World Terrain can create buildings, roads, trees, grass, and rivers based on Open Street Map data.

Real World Terrain is incredibly powerful and flexible. It has a powerful Editor API to automate the generation of terrains, and Runtime API positioning objects by coordinates, etc.

Real World Terrain has integration with the best assets of the Asset Store, which gives almost unlimited possibilities in the generation of terrains.

Features:

- Unity 2017.4 LTS and higher;
- Elevation Heightmaps:
    - ArcGIS resolution of max 10 meter per pixel;
    - Bing Maps resolution of max 10 meter per pixel;
    - Mapbox;
    - SRTM v4.1 resolution of 90 meters per pixel;
    - SRTM30 resolution of 30 meters per pixel.
- Texture providers: ArcGIS, DigitalGlobe, Map Quest, Mapbox, Mapy.CZ, Nokia Maps (here.com), Virtual Earth (Bing Maps), Open Street Map , Sentinel-2 + ability to download tiles from custom url;
- Satellite images resolution of max 0.25 meter per pixel;
- Can create: Unity Terrains, Meshes, Gaia stamps, RAW files;
- Can generate regular textures and Huge Texture (up to 26624x26624px) for each terrain.
- Can generate Terrain Layers based on Mapbox vector tile data;
- Can create objects based on Open Street Map:
    - Editable roads for EasyRoads 3D v3 and Road Architect;
    - Editable buildings for BuildR2 (editable) or built-in building engine;
    - Rivers;
    - Trees;
    - Grass.
- A tool to select area directly on Google Maps;
- A lot of extra tools for working with coordinates, objects and postprocessing;
- Unlimited number of generated terrains;
- Integration (optional): BuildR, EasyRoads 3D v3, Gaia, Gaia Pro, Huge Texture, Online Maps, Playmaker, Relief Terrain Pack, Road Architect, Vegetation Studio, Vegetation Studio Pro, VolumeGrass, WorldStreamer.

Terrain Quality Manager

https://assetstore.unity.com/packages/tools/terrain/terrain-quality-manager-28949

Terrain Quality Manager allows you to change the resolution of Heightmap, Detailmap, Alphamap and Basemap, **without losing data**.

If you need to increase the quality of terrain, or optimize terrain for better performance, with this tool you can do it.

Features:

- Unity v5.2 and higher;
- Allows you to change resolution of Heightmap, Detailmap, Alphamap, Basemap and Resolution Per Patch, without losing data;
- Works for single terrain, all terrains in scene, all terrains in project;
- Allows you to scale up and scale down terrain maps;
- Seamless result, when working with multiple terrains;
- Very easy to use.

uContext (FREE, PRO)

uContext Basic (FREE):

https://assetstore.unity.com/packages/tools/utilities/ucontext-basic-182221

uContext PRO (PAID):

https://assetstore.unity.com/packages/tools/utilities/ucontext-pro-141831

uContext is Editor PowerPack that take your workflow in Unity Editor to a next level, simplifies working with content, adds new features to the editor, corrects and improves the editor's built-in behaviors.
Over 40 useful tools for everyone in one asset.
Make your usual actions with lightning speed with uContext.

Key features:

- **Universal context menu** which is available everywhere in Unity Editor and allows you to quickly open the most useful windows, open components, create objects, add components, navigate and perform many more useful actions.

- **WAILA** (What Am I Looking At) + **Smart Selection**. WAILA displays one or all GameObjects under the cursor, and allows you to quickly find any object, even if it is hidden behind another object.
Smart Selection allows you to quickly select any object that WAILA shows.

- **Object Placer** allows you to easily and quickly create objects in the scene. Just point to the place where you want to create the object and press CTRL + SHIFT + Right Click.

- Ability to **open any component in a separate window**. The component will be displayed as if it were part of an inspector. Custom editors and third-party assets that modify the inspector (like Odin - Inspector and Serializer) are supported. It has a debug mode where only serialized fields will be displayed. The component will always be available, even if GameObject has lost focus.

- **View Gallery** - displaying images from all cameras in the scene and all View States, with the ability to set the view from the camera or View State with one click.
In addition, you can save View States for selection, and quickly restore it.

- **Quick Preview** shows the image from cameras and View States in Scene View.
Allows you to quickly view the scene from different angles, and adjust the scene view in one click.

- **Bookmarks** for any objects in a scene or project, allow you to always keep your most needed objects at hand.
In addition, you can store folders from the project in bookmarks, which gives you almost limitless possibilities for organizing content.

- The ability to quickly switch between open windows, even when the current window is maximized.

- **Recent windows** allows you to quickly open windows that are not closed recently.

- **Favorite Windows** allows you to create your own list of windows that are most useful to you and open them with one click. Built-in Unity Editor windows and third party asset windows are supported.

- **Smart Search** for GameObjects and Components in the scene, assets in the project, and items in the menu.
Extra search features:
Fuzzy search: allows you to search by characters and parts of a word, for example, enter «pw» to find «Project Window».
Search by type: Enter «:» and the type of object (in whole or in part) to search by type. For example, «wa:te» will find «08. Waila» texture in the project, or «DL:go» will find «Directional Light» GameObject in the scene.
Drag and Drop: You can drag and drop objects from search results into a scene or inspector.

- **Selection History** - uContext stores a history of object selections in the scene and project, and you can quickly switch between entries.

- **Duplicate Tool** - just drag GameObjects in the desired direction to create a duplicate. Ability to temporarily switch to the Duplicate Tool from the Move Tool or Transform Tool by pressing CTRL + SHIFT.

Additional features:
- Components in Hierarchy + display components errors and exceptions on Hierarchy;
- Align & Distribute Tools;
- Drop To Floor + Advanced Drop To Floor;
- Grouping and ungrouping GameObjects;
- Quick replacement GameObjects;
- Recent Scenes;
- Displaying position, rotation and scale in Scene View when using the appropriate tools;
- Transform Editor Tools;
- Displaying the size of selected objects;
- Tool for calculating distances in the scene;
- Fast switching between Scene View and Game View (CTRL + SHIFT + TAB);
- Auto switch to maximized Game View on enter playmode when Scene View is maximized;
- Temporary objects. Allows you to create temporary objects that will be destroyed at start the scene;
- Fast zoom in / out scene view;
- Rename GameObjects in Scene View by F2;
- Integration with Online Maps, Real World Terrain;
- Built-in update system.

Improves the built-in behavior of the Unity Editor:
- Drag and drop objects onto Canvas;
- Add Component by shortcut;
- Change the size of the brush Terrain Editor using the mouse wheel;
- Maximize Game View in playmode by SHIFT + Space;
- Maximize any window by F11.

uContext is fully customizable. If a feature does not fit your workflow, you can adjust or disable this feature.

Views

https://assetstore.unity.com/packages/tools/utilities/views-168760

Views is a standalone version of the View Gallery and Quick Preview features from ***uContext***, allowing you to save, view and restore View States, and preview cameras.

View Gallery displaying images from all cameras in the scene and all View States, with the ability to set the view from the camera or View State with one click.

Quick Preview shows the image from cameras and View States in Scene View. Allows you to quickly view the scene from different angles, and adjust the scene view in one click.

View State is a Scene View state. Something like a visual bookmark.

Final words

We sincerely hope that you will enjoy using uPano.

If you have any questions or problems, please contact us.

We will try to help you as soon as possible.

Please do not forget to add your review in Unity Asset Store.

It is very important for us to have feedback from users to make our asset better.

This will help other users to create the right understanding of the asset quality and features.

Links

Product page: https://infinity-code.com/assets/upano

API Reference: https://infinity-code.com/docs/api/upano/

Email: support@infinity-code.com

Forum: https://forum.infinity-code.com

YouTube: https://www.youtube.com/playlist?list=PL2QU1uhBMew87wEKLcaTRihCft-buVmdy

Vimeo: https://vimeo.com/channels/upano

Twitter: https://twitter.com/InfinityCodeCom