

Documentation for Image and 360 Image Viewer asset

Not native gallery for photos and panoramas that you upload into your app.

360 Image Viewer is a gallery that aims to save your time making your own gallery viewer. Current version provides you gallery, photo and panorama viewers.

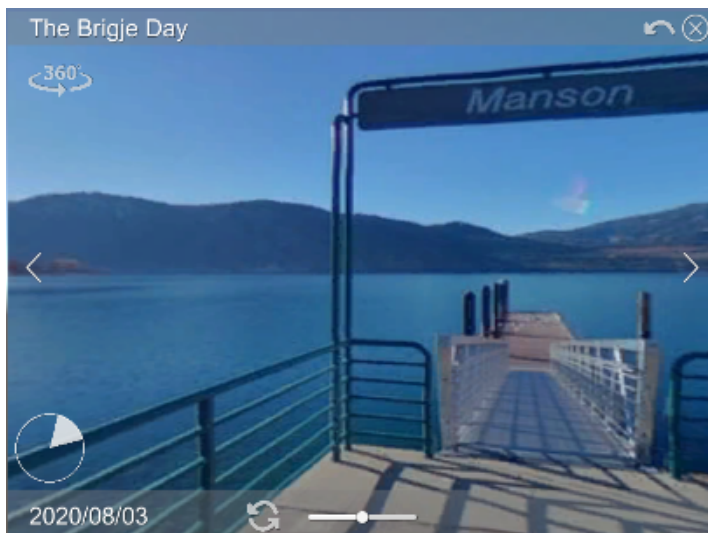
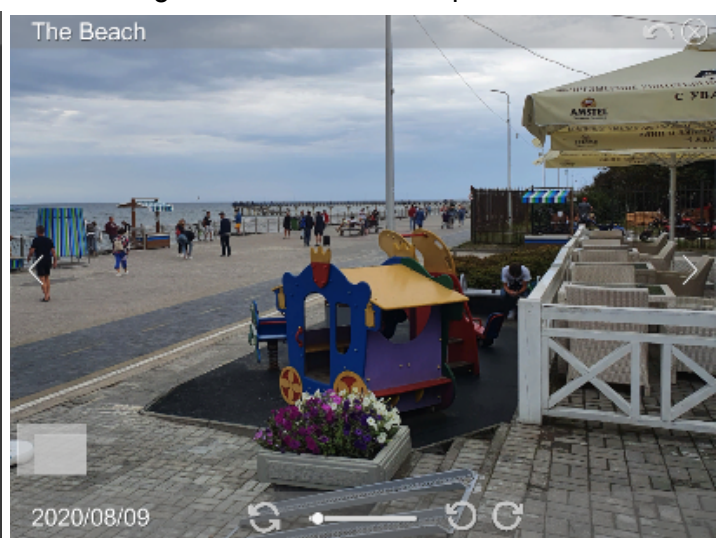
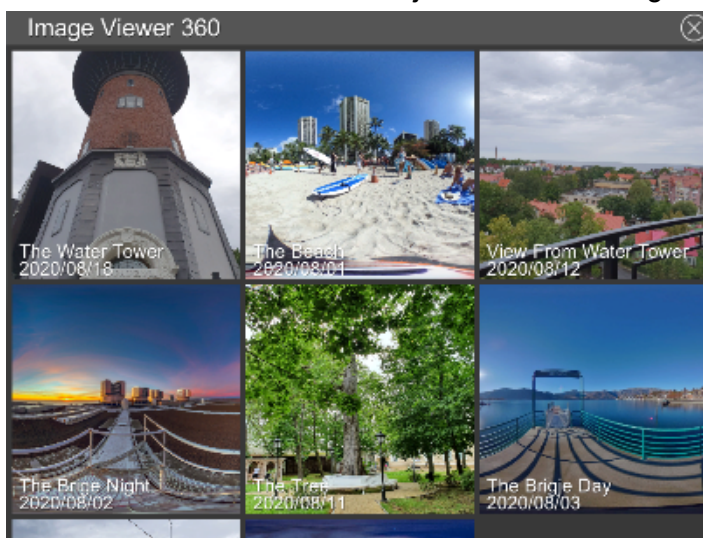
The code base doesn't have any dependencies in code and is made in unity version 2017.4 LTS.

With small effort the prefabs can be rebuilt in older Unity versions.

The asset shows and is able to switch between a list of images and 360 images (panorama).

The asset source code is open and can be customized according to your needs.

The asset contains two main objects - one for images and 360 images. Both have some special features.



Features

Gallery View :

- Shows all images in scrollview as tiles.

Images View :

- Shows standard photo images.
- Zoom in and Zoom out images.
- Rotates image by 90 degrees to left and right.
- Move image with swipes according to swipe direction, when image zoomed.
- Reset button can reset zoom and rotation.
- Image map shows viewport position relative to image. That is suitable when the image zoomed and we may need to know what part of the image is seen.

360 Image View :

- Shows full 360 photo images (panorama).
- Zoom in and Zoom out 360 images.
- Move image with swipes according to swipe direction allowing to observe all parts of the 360 image.
- Reset button can reset zoom and rotation.
- Image map shows viewport position relative to image. That is suitable when the 360 image zoomed and/or rotated we may need to know what part of the 360 image is seen.

After importing the asset you will find all the Viewer files at Assets\PhotoViewer folder.

The folder structure:

Assets\PhotoViewer\Images	graphic for asset ui
Assets\PhotoViewer\Photos	Demo standard images
Assets\PhotoViewer\Photos360	Demo 360 images
Assets\PhotoViewer\Prefabs	Asset prefabs
Assets\PhotoViewer\Scenes	Example scene
Assets\PhotoViewer\Scripts	Asset code base
Assets\PhotoViewer\Scripts\Buttons	Scripts for buttons
Assets\PhotoViewer\Scripts\Panorama	Codebase related to 360 images view
Assets\PhotoViewer\Scripts\Photo	Codebase related to standard images view

The main class of the asset is PhotoViewer.cs . It receives images as List of ImageData or ImageDate Type. The type contains Sprite and two strings fields - one for image name and other for image date. The example:

<pre> public ImageData Data; public List<ImageData> Datas; public PhotoViewer Viewer; Viewer.AddImageData(Data); Viewer.AddImageData(Datas); Viewer.Show(); </pre>	<pre> [Serializable] public struct ImageData { public Sprite Sprite; public string Name; public string Date; } </pre>
---	---

PhotoViewer contains:

Fields

CloseImageViewer	public event perform on viewer closed
------------------	---------------------------------------

Serialised fields

Gallery _gallery	Gallery View - viewer structure object in prefab hierarchy. Shows all images as scrollable set of tales
Panorama _panorama	360 Image View - viewer structure object in prefab hierarchy. Shows 360 images
Photo _photo	Image view - viewer structure object in prefab hierarchy. Shows standard images
GameObject _btnReturn	The button to return from Image and Image 360 view into Gallery view

Public Methods

AddImageData(ImageData)	Add information for single image
AddImageData(List<ImageData>)	Add list of information for images
CloseViewer	Close image viewer
Clear	Clear image viewer from all added data and reset field
Show	Show first image in the list
NextImage	Show next image from the list
PrevImage	Show previous image from the list

For example, reference to TestImageLoader.cs where images are loaded from scriptable object.

The PhotoViewer.cs class has links to three main objects in its prefab structure. Those objects are Gallery to see all loaded images as a tiled scrollable list, Photo.cs for standard images and Panorama.cs for 360 images.

Photo and Panorama classes share some logic and are inherited from AbstractView.cs class. When ImageViewer realizes that the next image is a standard image, it disables PanoramaView object, enables ImageView object and initiates with ImageData.

AbstractView.cs

Serialized fields

Text _name	UI object in prefab structure to show name on an image
Text _date	UI object in prefab structure to show image date
Slider _zoomSlider	UI object in prefab structure to zoom in/out image
ResetButton _btnReset	UI object in prefab structure to reset change done to an image

Methods

ApplyInput	Adds input delta position for moving photo/rotating panorama
ZoomSlider	The slider method to zoom on view
Show	Shows selected imageData on view
Reset	Reset changes (rotate/move/zoom) to a view

Photo.cs

Serialized fields

Image _image	UI element in prefab structure to show standard image
Sprite _defaultImage	Default image if no image was load
PhotoMap _photoMap	Element in prefab structure to show image map with relatively sized image and viewport

Public Methods

RotateLeft	Rotate image object counter clockwise
RotateRight	Rotate image object clockwise
ApplyInput	Moves scaled image
Clear	Clears view data (rotation, image, scale)

Panorama.cs

Serialized fields

PanoramaRotator _panoramaCameraPrefab	Camera Prefab to render panorama image
GameObject _panoramaSpherePrefab	Sphere with flipped normals inside. Use image as material texture
PanoramaMap _panoramaMap	Element in prefab structure to show image map with relatively sized image and viewport
Image _icon360	UI Image from prefab structure to show notification that is panorama
bool _animateIcon360	Animate 360 icon on image show when true

Public Methods

Clear	Clears view data (rotation, image, zoom)
-------	--

Gallery.cs

Serialized fields

Text _name	UI object in prefab structure to show name of the gallery
RectTransform _content	UI object in prefab structure where tiles will be instantiated
GalleryTile _galleryTilePrefab	Prefab of a gallery tale

Public Methods

Init	Initialize the gallery with Image Datas
Clear	Clear the gallery from created tales in release the data