VR Gallery

Generated by Doxygen 1.9.4

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 AppState Class Reference

Static app state class.

Static Public Attributes

- static bool **IsGalleryRendered** = false
- static Artwork[ ] **CurrentArtworkslist**
- static List< string > **FavoriteFilters**
- static Artwork[ ] **AllArtworks**
- static string[ ] **AllTags**

3.1.1 Detailed Description

Static app state class.

This class is a static class that holds all the data that is used throughout the application

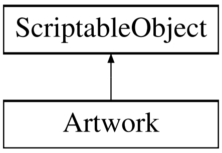
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/AppState.cs

3.2 Artwork Class Reference

Artwork scriptable object.

Inheritance diagram for Artwork:



Public Attributes

- Sprite **image**
- new string **name**
- string **author**
- string **description**
- int **month**
- int **year**
- string[ ] **tags**
- AudioClip **audio**

3.2.1 Detailed Description

Artwork scriptable object.

It's a class that contains all the information about an artwork

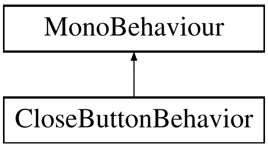
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/Artwork.cs

3.3 CloseButtonBehavior Class Reference

Behavior for close button.

Inheritance diagram for CloseButtonBehavior:



Public Attributes

- GameObject **MenuCanvas**

Private Member Functions

- void Start ()
- void **TaskOnClick** ()

  If the gallery is rendered, close the menu and return to the gallery. If not, close the game.

3.3.1 Detailed Description

Behavior for close button.

It's a button that closes the game or the menu, depending on the state of the game

3.3.2 Member Function Documentation

3.3.2.1 Start()

```
void CloseButtonBehavior.Start ( )  [private]
```

The Start() function is called when the script is first run. It adds a listener to the button that calls the TaskOnClick() function when the button is clicked

The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/CloseButtonBehavior.cs

3.4 Filters Class Reference

Filter artwors.

Static Public Member Functions

- static string[ ] GetAllTags (Artwork[ ] artworks)
- static string[ ] GetAllAuthors (Artwork[ ] artworks)
- static List< string > GetAllYears (Artwork[ ] artworks)
- static Artwork[ ] FilterForTag (string tag, Artwork[ ] artworks)
- static Artwork[ ] FilterForAuthor (string author, Artwork[ ] artworks)
- static Artwork[ ] FilterForCv (string author, Artwork[ ] artworks)
- static Artwork[ ] FilterForYear (int year, Artwork[ ] artworks)
- static ? Artwork GetArtworkByName (string name, Artwork[ ] artworks)

Static Public Attributes

- static List< Artwork > **SelectedArtworkslist**

3.4.1 Detailed Description

Filter artwors.

It's a class that contains methods that filter an array of Artwork objects based on different criteria

3.4.2 Member Function Documentation

3.4.2.1 FilterForAuthor()

```
static Artwork[ ] Filters.FilterForAuthor (
             string author,
             Artwork[ ] artworks )  [static]
```

It takes an array of Artwork objects and a string, and returns an array of Artwork objects that have the same author as the string

Parameters

author	The author to filter for
artworks	The array of artworks to filter

Returns

An array of Artwork objects.

3.4.2.2 FilterForCv()

```
static Artwork[] Filters.FilterForCv (
            string author,
            Artwork[] artworks )  [static]
```

It takes an author and an array of artworks, filters the array for the author, and then orders the array by month and year

Parameters

author	
artworks	an array of Artwork objects

Returns

An array of Artwork objects.

3.4.2.3 FilterForTag()

```
static Artwork[] Filters.FilterForTag (
            string tag,
            Artwork[] artworks )  [static]
```

It takes an array of artworks and a tag, and returns an array of artworks that have that tag

Parameters

tag	The tag to filter for
artworks	The array of Artwork objects to filter

Returns

An array of Artwork objects.

3.4.2.4 FilterForYear()

```
static Artwork[] Filters.FilterForYear (
             int year,
             Artwork[] artworks )  [static]
```

It takes an array of Artwork objects and returns an array of Artwork objects that were created in the year specified by the year parameter

Parameters

year	The year to filter for
artworks	The array of Artwork objects to filter.

Returns

An array of Artwork objects.

3.4.2.5 GetAllAuthors()

```
static string[] Filters.GetAllAuthors (
             Artwork[] artworks )  [static]
```

It takes an array of Artwork objects, loops through each one, and adds the author to a list if it's not already in the list

Parameters

artworks	An array of Artwork objects.

Returns

An array of strings

3.4.2.6 GetAllTags()

```
static string[] Filters.GetAllTags (
             Artwork[] artworks )  [static]
```

It takes an array of Artwork objects, loops through each Artwork object, loops through each tag in the Artwork object, and adds the tag to a list of tags if it doesn't already exist in the list

Parameters

artworks	The array of artworks to get the tags from.

Returns

An array of strings.

3.4.2.7 GetAllYears()

```
static List< string > Filters.GetAllYears (
              Artwork[] artworks )  [static]
```

It takes an array of Artwork objects, loops through each one, and adds the year to a list if it's not already in the list

Parameters

artworks	The array of Artwork objects

Returns

A list of strings

3.4.2.8 GetArtworkByName()

```
static ?  Artwork Filters.GetArtworkByName (
              string name,
              Artwork[] artworks )  [static]
```

It takes a string and an array of Artwork objects, and returns the first Artwork object in the array whose name property matches the string

Parameters

name	The name of the artwork you want to get.
artworks	The array of Artwork objects that you want to search through.

Returns

The first artwork with the name that matches the name parameter.

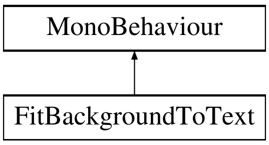
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/Filters.cs

3.5 FitBackgroundToText Class Reference

Fits background to text.

Inheritance diagram for FitBackgroundToText:



Public Attributes

- RectTransform **textRect**

Private Member Functions

- void OnEnable ()

Private Attributes

- float **padding** = 10

3.5.1 Detailed Description

Fits background to text.

It takes the height of the text and adds a padding to it, then sets the height of the background to the new height.

3.5.2 Member Function Documentation

3.5.2.1 OnEnable()

```
void FitBackgroundToText.OnEnable ( )  [private]
```

When the object is enabled, get the height of the text and add the padding to it, then set the height of the object to that value.

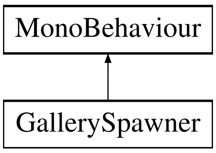
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/FitBackgroundToText.cs

3.6 GallerySpawner Class Reference

Spawns a gallery.

Inheritance diagram for GallerySpawner:



Public Member Functions

- void **CountGalleryLength** ()

  It counts the number of artworks.
- void SpawnGallery (string style)
- void SetStyle (string style)

Public Attributes

- GameObject **WallWithoutPaintingClassic**
- GameObject **WallWithoutPaintingWood**
- GameObject **WallWithoutPaintingBricks**
- GameObject **classic**
- GameObject **wood**
- GameObject **bricks**
- Transform **TPArea**
- float **size** = 4

Private Member Functions

- void **Spawn** ()

  It instantiates a prefab with a painting on it.
- void **SetArtwork** ()

  It sets the image, text, and audio of the artwork.

Private Attributes

- GameObject **prefab**
- GameObject **wallWithoutPainting**
- Artwork[ ] **ArtworkPool**
- int **artworkCount**
- Artwork **artwork**
- GameObject **currentPrefab**
- Image **image**
- TextMeshProUGUI **infoText**
- GuideVoice **guidevoice**
- int **index**

3.6.1 Detailed Description

Spawns a gallery.

It spawns a gallery of paintings, each of which has an image, a name, an author, a year, a description and an audio clip.

3.6.2 Member Function Documentation

3.6.2.1 SetStyle()

```
void GallerySpawner.SetStyle (
             string style )
```

This function is called when the user selects a style from the style menu. It sets the prefab variable to the prefab of the selected style and sets the wallWithoutPainting variable to the wallWithoutPainting prefab of the selected style

Parameters

style	the style of the wall

3.6.2.2 SpawnGallery()

```
void GallerySpawner.SpawnGallery (
             string style )
```

It takes a string as an argument, sets the style of the gallery, counts the length of the gallery, and then spawns the gallery.

Parameters

style	The style of the gallery.

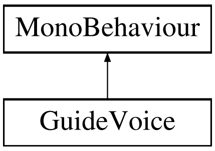
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/GallerySpawner.cs

3.7 GuideVoice Class Reference

handels voice activation for artwork description

Inheritance diagram for GuideVoice:



Public Types

- enum ActivationMode { **ButtonTriggerd** , **ProximityTriggerd** }

Public Member Functions

- void **ToggleVoice** ()

  toogels guide voice

Public Attributes

- AudioClip **audioClip**

  audioclip to be played

Properties

- ActivationMode **activationMode** `[get, set]`

  prepares for new interaction based on choosen ActivationMode
- bool **isSpeaking** `[get, set]`

  sets of image sprite based on if the audioclip is playing

Private Member Functions

- void **Start** ()

  initialize sprites, sets activationMode, sets audioclip
- void **Update** ()

  updates isSpeaking, based on audioSource.isPlaying, to be processed further
- void OnTriggerEnter (Collider other)
- void OnTriggerExit (Collider other)

Private Attributes

- ActivationMode **_activationMode** = ActivationMode.ButtonTriggerd

  current ActivationMode
- bool **_isSpeaking** = false

  is the audioclip currently playing
- AudioSource **audioSource**

  audiosource to be used
- Sprite **playArrowSprite**

  holds playArrowSprite
- Sprite **stopSprite**

  holds stopSprite

3.7.1 Detailed Description

handels voice activation for artwork description

Plays an audioclip with the artwork description

3.7.2 Member Enumeration Documentation

3.7.2.1 ActivationMode

```
enum GuideVoice.ActivationMode
```

ButtonTriggerd can be used when you want the user to press a button to play the audioclip, ProximityTriggerd is used when you want to play the clip if the user comes near the artwork

3.7.3 Member Function Documentation

3.7.3.1 OnTriggerEnter()

```
void GuideVoice.OnTriggerEnter (
             Collider other )  [private]
```

starts to play the audioclip if collides with MainCamera

Parameters

other	objects which this is colliding with

3.7.3.2 OnTriggerExit()

```
void GuideVoice.OnTriggerExit (
             Collider other )  [private]
```

stops to play the audioclip if collider exiting is MainCamera

Parameters

objects	which this is colliding with

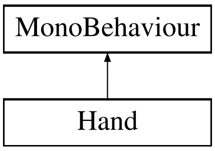
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/GuideVoice.cs

3.8 Hand Class Reference

Animate hand.

Inheritance diagram for Hand:



Public Attributes

- float **speed**

Private Member Functions

- void Start ()
- void Update ()
- void AnimateHand ()

Private Attributes

- Animator **animator**
- float **gripTarget**
- float **triggerTarget**
- float **gripCurrent**
- float **triggerCurrent**

3.8.1 Detailed Description

Animate hand.

It's a class that animates a hand model

3.8.2 Member Function Documentation

3.8.2.1 AnimateHand()

```
void Hand.AnimateHand ( )  [private]
```

If the current grip value is not equal to the target grip value, then move the current grip value towards the target grip value by a certain amount each frame

3.8.2.2 Start()

```
void Hand.Start ( )  [private]
```

The function is called when the script is first run. It gets the Animator component from the GameObject that the script is attached to

3.8.2.3 Update()

```
void Hand.Update ( )  [private]
```

The function is called every frame. It checks if the hand is open or closed, and if it is open, it checks if the hand is moving. If the hand is moving, it plays the animation for the hand moving. If the hand is not moving, it plays the animation for the hand being open. If the hand is closed, it plays the animation for the hand being closed

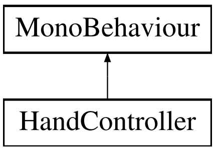
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/Hand.cs

3.9 HandController Class Reference

Hand Controller.

Inheritance diagram for HandController:



Public Attributes

- Hand **hand**

Private Member Functions

- void Start ()
- void Update ()

Private Attributes

- ActionBasedController **actionBasedController**

3.9.1 Detailed Description

Hand Controller.

This class is responsible for reading the values of the grip and trigger actions and passing them to the hand class

3.9.2 Member Function Documentation

3.9.2.1 Start()

```
void HandController.Start ( )  [private]
```

It gets the ActionBasedController component from the game object that this script is attached to

3.9.2.2 Update()

```
void HandController.Update ( )  [private]
```

The function is called every frame and it reads the value of the grip and trigger actions and passes them to the hand object

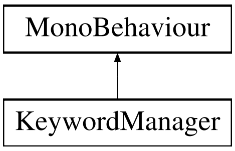
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/HandController.cs

3.10 KeywordManager Class Reference

Manager for keywords and filters.

Inheritance diagram for KeywordManager:



Public Member Functions

- void GetArtworkByName (string name)
- void FilterForYear (int year)
- void FilterForCv (string author)
- void FilterForTag (List< string > tags)
- void **CountArtworks** ()

  It counts the number of artworks in the ArtworkPool and displays it in the ArtworkCountField.
- void CreateFilterButtons ()
- void CreateAuthorButtons ()
- void AddToFilterList (string filter)
- void **SaveActualArtworkpool** ()

  It saves the current artworks list to the AppState.CurrentArtworkslist variable.
- void ResetPool ()

Public Attributes

- string[ ] **AllTags**
- string[ ] **AllAuthors**
- Artwork[ ] **ArtworkPool**
- TextMeshProUGUI **ArtworkCountField**
- TextMeshProUGUI **CurrentFilter**
- TextMeshProUGUI **CurrentAuthor**
- Transform **FilterButtonsPanel**
- Transform **AuthorButtonsPanel**
- Toggle **FilterButon**
- Button **CVButon**
- GameObject **AlarmPanel**
- GameObject **ProfileManager**

Private Member Functions

- void Start ()
- bool IsTagsListEmpty ()
- void DisplayFilters ()

Private Attributes

- Artwork[ ] **AllArtworks**
- List< string > **tagsList**

3.10.1 Detailed Description

Manager for keywords and filters.

It's a class that manages the keywords and filters for the artworks

3.10.2 Member Function Documentation

3.10.2.1 AddToFilterList()

```
void KeywordManager.AddToFilterList (
             string filter )
```

If the filter is in the list of all tags, then add it to the list of filters. If it's not in the list of all tags, then display an alarm panel

Parameters

filter	the string that is being added to the list

3.10.2.2 CreateAuthorButtons()

```
void KeywordManager.CreateAuthorButtons ( )
```

For each author in the list of all authors, create a button, set the button's label to the author's name, and add a listener to the button that calls the function FilterForCv() when the button is clicked

3.10.2.3 CreateFilterButtons()

```
void KeywordManager.CreateFilterButtons ( )
```

It creates a toggle button for each tag in the AllTags list, and adds a listener to each button that adds the button's text to the filter list when the button is toggled on.

3.10.2.4 DisplayFilters()

```
void KeywordManager.DisplayFilters ( )  [private]
```

If the list of tags is empty, display all artworks. If the list of tags is not empty, filter the artworks for the tags in the list.

3.10.2.5 FilterForCv()

```
void KeywordManager.FilterForCv (
             string author )
```

It takes a string as an argument, filters the list of artworks for the given author and then sets the text of the current author text field to the given author

Parameters

author	The name of the author to filter for

3.10.2.6 FilterForTag()

```
void KeywordManager.FilterForTag (
             List< string > tags )
```

If there's only one tag, then filter the artworks for that tag. If there's more than one tag, then filter the artworks for each tag and add them to a list. If there's no tags, then set the artwork pool to null

Parameters

tags	A list of tags to filter for.

3.10.2.7 FilterForYear()

```
void KeywordManager.FilterForYear (
             int year )
```

It takes a year as an argument, and then filters the ArtworkPool to only contain artworks from that year

Parameters

year	The year to filter for

3.10.2.8 GetArtworkByName()

```
void KeywordManager.GetArtworkByName (
             string name )
```

If the artwork is found, it is added to the ArtworkPool. If it is not found, the ArtworkPool is emptied

Parameters

name	The name of the artwork to search for

3.10.2.9 IsTagsListEmpty()

```
bool KeywordManager.IsTagsListEmpty ( )  [private]
```

If the tagsList is empty, return true, else return false

Returns

The method returns a boolean value.

3.10.2.10 ResetPool()

```
void KeywordManager.ResetPool ( )
```

It resets the pool of artworks to the full list of artworks, resets the filter and author text and counts the artworks

3.10.2.11 Start()

```
void KeywordManager.Start ( )  [private]
```

The function is called when the application starts. It resets the pool of artworks, creates a list of tags, gets all the tags from the artworks, gets all the authors from the artworks, creates the filter buttons, and creates the author buttons

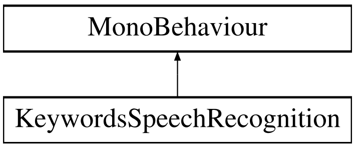
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/KeywordManager.cs

3.11 KeywordsSpeechRecognition Class Reference

Speech recognition class.

Inheritance diagram for KeywordsSpeechRecognition:



Public Member Functions

- void **StartSpeechToText** ()

    It starts the speech to text function.
- void StopSpeechToText ()
- string[ ] ReturnGuesses ()

Public Attributes

- string[ ] **taglist**
- string **firstGuess**
- List< string > **secondGuess**
- GameObject **FilterManager**
- Toggle **MicroToggle**
- TextMeshProUGUI **ToggleText**

Private Member Functions

- void **Start** ()

    It adds a listener to the MicroToggle object.
- void ToggleValueChanged (Toggle change)
- IEnumerator **WaitAndReturnText** ()

    It waits for 1 second, then it adds the first guess to the filter list.
- void OnEnable ()
- void DictationRecognizer_DictationHypothesis (string text)
- void DictationRecognizer_DictationResult (string text, ConfidenceLevel confidence)
- void DictationRecognizer_DictationError (string error, int hresult)
- void OnDisable ()

Private Attributes

- DictationRecognizer **dr**

3.11.1 Detailed Description

Speech recognition class.

It listens to what you say and tries to find a match in the taglist. If it finds a match it adds it to the firstGuess or secondGuess list

3.11.2 Member Function Documentation

3.11.2.1 DictationRecognizer_DictationError()

```
void KeywordsSpeechRecognition.DictationRecognizer_DictationError (
            string error,
            int hresult )  [private]
```

The function is called when the dictation recognizer encounters an error

Parameters

error	The error message.
hresult	The error code.

3.11.2.2 DictationRecognizer_DictationHypothesis()

```
void KeywordsSpeechRecognition.DictationRecognizer_DictationHypothesis (
            string text )  [private]
```

The function takes the text that the user is currently saying and splits it into a list of words. It then adds the entire text to the list. It then checks if any of the words in the list are in the list of tags. If they are, it adds them to a list of second guesses

Parameters

text	The text that was recognized by the dictation recognizer.

3.11.2.3 DictationRecognizer_DictationResult()

```
void KeywordsSpeechRecognition.DictationRecognizer_DictationResult (
            string text,
            ConfidenceLevel confidence )  [private]
```

If the user says a word that is in the taglist, then add it to the firstGuess variable. If the firstGuess variable is not empty, then add the firstGuess variable to the secondGuess list. Then, set the firstGuess variable to the word that the user just said

Parameters

text	The text that was recognized.
ConfidenceLevel	The confidence level of the dictation recognizer.

3.11.2.4 OnDisable()

```
void KeywordsSpeechRecognition.OnDisable ( )  [private]
```

The OnDisable function is called when the script is disabled.

dr.Status == SpeechSystemStatus.Running checks if the speech recognizer is running.

dr.Stop() stops the speech recognizer.

dr.DictationResult -= DictationRecognizer_DictationResult removes the event handler for the DictationResult event.

dr.DictationError -= DictationRecognizer_DictationError removes the event handler for the DictationError event.

dr.Dispose() disposes of the speech recognizer.

3.11.2.5 OnEnable()

```
void KeywordsSpeechRecognition.OnEnable ( )  [private]
```

The OnEnable function is called when the script is enabled. It creates a new list of strings called second↩Guess, creates a new DictationRecognizer called dr, and adds the DictationRecognizer_DictationResult, Dictation↩Recognizer_DictationHypothesis, and DictationRecognizer_DictationError functions to the dr.DictationResult, dr.↩DictationHypothesis, and dr.DictationError events.

3.11.2.6 ReturnGuesses()

```
string[] KeywordsSpeechRecognition.ReturnGuesses ( )
```

It returns an array of strings where the first string is the most likely tag and the rest of the strings are the other possible tags

Returns

An array of strings.

3.11.2.7 StopSpeechToText()

```
void KeywordsSpeechRecognition.StopSpeechToText ( )
```

It stops the speech recognition, and then calls a coroutine that waits for a second before returning the text

3.11.2.8 ToggleValueChanged()

```
void KeywordsSpeechRecognition.ToggleValueChanged (
            Toggle change )  [private]
```

If the toggle is on, start the speech to text function and change the text to "Ein Tag aussprechen". If the toggle is off, stop the speech to text function and change the text to "zum Sprechen klicken"

Parameters

Toggle	The toggle that was changed.

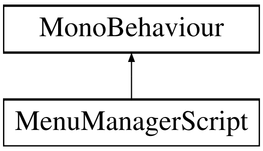
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/KeywordsSpeechRecognition.cs

3.12 MenuManagerScript Class Reference

Controls the tutorial-menu.

Inheritance diagram for MenuManagerScript:



Public Member Functions

- void **NextSite** ()
  It navigates to the next tutorial site.
- void **PreviousSite** ()
  It navigates to the previous tutorial site.

Public Attributes

- int **totalTutorialSites**
  indicates how many tutorial sites there are and which one is currently visible
- int **currentSite**
- GameObject **firstPanel**
  for the different tutorial sites
- GameObject **secondPanel**

Private Member Functions

- void Start ()

Private Attributes

- GameObject[ ] **_panels**

3.12.1 Detailed Description

Controls the tutorial-menu.

It provides the tutorial navigation, including next site & previous site

3.12.2 Member Function Documentation

3.12.2.1 Start()

```
void MenuManagerScript.Start ( )  [private]
```

The function is called when the application starts. It sets the currently visible site and how many tutorial sites there are. Furthermore it fills an array with the panels for each tutorial site.

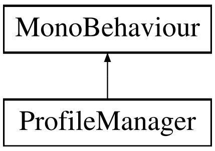
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/MenuManagerScript.cs

3.13 ProfileManager Class Reference

Manages the profile data.

Inheritance diagram for ProfileManager:



Public Member Functions

- void IncreaseStyleFrequency (string name)
- void Sort (int[ ] arr)
- void ShowProfile ()
- void **HideProfile** ()

  Hides the profile by deactivating the profile panel.
- void GetLastTags (List< string > tagsList)
- void **SavePrefs** ()

  Saves every tag with its frequency in the Player-Prefs.
- void **LoadPrefs** ()

  Loads the tags from the Player-Prefs.

Public Attributes

- GameObject **galleryStyle**
- GameObject **profilePanel**
- GameObject **tagText1**
- GameObject **tagText2**
- List< Sprite > **galleryStyles**
- string[ ] **tags**
- int[ ] **tagsFrequency**

Static Public Attributes

- static ProfileManager **pManager**

  to call methods from another script

Private Member Functions

- void Start ()
- IEnumerator **WaitAndGetTags** ()
- void **InitDictionaries** ()

  It initialize the dictionary for the different gallery styles and the corresponding sprites.
- void **ChangeSprite** (Sprite sprite)
- void **InitializeTags** ()
- void **DetectFavTags** ()

Private Attributes

- bool **_visible**
- Dictionary< string, int > **_styleDictionary**
- Dictionary< string, Sprite > **_styleSpritesDictionary**
- Image **_imageRenderer**
- TextMeshProUGUI **_text1Comp**
- TextMeshProUGUI **_text2Comp**
- GameObject **_profileButton**

3.13.1 Detailed Description

Manages the profile data.

It holds a list of tags and gallery styles with each corresponding frequency and manages the according preferences determined by the users input

3.13.2 Member Function Documentation

3.13.2.1 GetLastTags()

```
void ProfileManager.GetLastTags (
            List< string > tagsList )
```

It gets the last used tags in the gallery and increases the frequency for each viewed tag. Furthermore it saves it via the Player-Prefs

Parameters

tagsList	A string List that holds the last tags viewed by the user.

3.13.2.2 IncreaseStyleFrequency()

```
void ProfileManager.IncreaseStyleFrequency (
            string name )
```

Increases the frequency for a specific gallery style

Parameters

name	The name for which the frequency should be increased

3.13.2.3 ShowProfile()

```
void ProfileManager.ShowProfile ( )
```

Shows the profile panel by activating it. Meanwhile it detects the favorite tags and sets it automatically in the profile.

3.13.2.4 Sort()

```
void ProfileManager.Sort (
             int[] arr )
```

Sorts the given integer array. Highest is at index 0, lowest at the end of the array

Parameters

arr	The integer array to be sorted

3.13.2.5 Start()

```
void ProfileManager.Start ( )  [private]
```

The function is called when the application starts. It sets all variables and initialize the dictionary containing every gallery style with its frequency. Starts the coroutine for getting all tags.

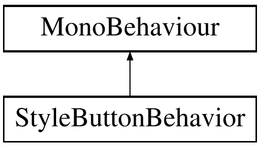
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/ProfileManager.cs

3.14 StyleButtonBehavior Class Reference

Behavior for style button.

Inheritance diagram for StyleButtonBehavior:



Public Attributes

- GameObject **SpawnManager**

Private Member Functions

- void Start ()
- void **TaskOnClick** ()

  When the user clicks on a task, the old galley is destroyed, the new gallery is spawned.
- void **DestroyGallery** ()

  It finds all the objects with the tag "gallery" and destroys them.

Private Attributes

- GameObject[ ] **oldGalleryPrefabs**

3.14.1 Detailed Description

Behavior for style button.

When button clicked, destroys the current gallery and spawns a new one

3.14.2 Member Function Documentation

3.14.2.1 Start()

```
void StyleButtonBehavior.Start ( )  [private]
```

The function is called when the script is first run. It adds a listener to the button that calls the TaskOnClick() function when the button is clicked

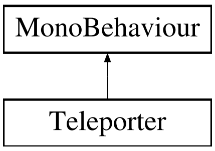
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/StyleButtonBehavior.cs

3.15 Teleporter Class Reference

Custom teleporter.

Inheritance diagram for Teleporter:



Public Attributes

- GameObject **TeleportRay**
- TeleportationArea **TPArea**

Private Member Functions

- void **Start** ()

  The function is called when the scene starts.
- void OnEnable ()
- void OnDisable ()
- void ButtonPressed (InputAction.CallbackContext obj)
- async void ButtonReleased (InputAction.CallbackContext obj)

Private Attributes

- InputActionReference **teleportInput**
- TeleportationProvider **teleportationProvider**

3.15.1 Detailed Description

Custom teleporter.

Teleporter, that activates ray only after pressing trigger button

3.15.2 Member Function Documentation

3.15.2.1 ButtonPressed()

```
void Teleporter.ButtonPressed (
             InputAction.CallbackContext obj )  [private]
```

When the button is pressed, the TeleportRay is set to active

Parameters

obj	The context of the action.

3.15.2.2 ButtonReleased()

```
async void Teleporter.ButtonReleased (
             InputAction.CallbackContext obj )  [private]
```

When the button is released, the raycast is turned off

Parameters

obj	The context of the action.

3.15.2.3 OnDisable()

```
void Teleporter.OnDisable ( ) [private]
```

When the script is disabled, disable the teleportInput action and remove the event listeners for the button pressed and button released events

3.15.2.4 OnEnable()

```
void Teleporter.OnEnable ( ) [private]
```

When the script is enabled, enable the teleportInput action, and add the ButtonPressed and ButtonReleased functions to the started and canceled events.

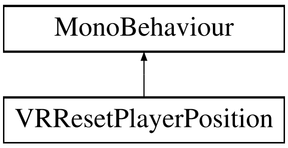
The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/VR/Teleporter.cs

3.16 VRResetPlayerPosition Class Reference

Reset player position.

Inheritance diagram for VRResetPlayerPosition:



Public Member Functions

- void **ResetPosition** ()

  The function resets the player's position and rotation.

Public Attributes

- Transform **ResetTransform**
- GameObject **Player**
- Camera **Camera**

3.16.1 Detailed Description

Reset player position.

It resets the player's position to the position of the reset transform

The documentation for this class was generated from the following file:

- C:/Users/kt/VRGallery/Assets/Scripts/VR/VRResetPlayerPosition.cs

Index