Class TargetHandler

Class for handling all actions concerning the targets in the environment.

Inheritance

↳ System.Object
  ↳ TargetHandler

Namespace: DocFx (DocFx.html)

Assembly: DocFx.dll

Syntax

```
public class TargetHandler : MonoBehaviour
```

Fields

currentTargetItems

List to store current targets.

Declaration

```
public List<TargetFacade> currentTargetItems
```

Field Value

Type	Description
System.Collections.Generic.List<TargetFacade (DocFx.TargetFacade.html)>	

item

Variable to store the current selected target.

Declaration

```
public string item
```

Field Value

Type	Description
System.String	

LoadURL

Address for loading the data to create target gameobjects.

Declaration

```
public const string LoadURL = "https://ar-app-rest-api.herokuapp.com/positions"
```

Field Value

Type	Description
System.String	

navigationController

Instance were the class is initialized.

Declaration

```
public NavigationController navigationController
```

Field Value

Type	Description
NavigationController (DocFx.NavigationController.html)	

navigationXOffset

Slider to change x location of selected target.

Declaration

```
public Slider navigationXOffset
```

Field Value

Type	Description
Slider	

navigationZOffset

Slider to change z location of selected target.

Declaration

```
public Slider navigationZOffset
```

Field Value

Type	Description
Slider	

NewTargetNameInput

Input field to give the new target added a name.

Declaration

```
public InputField NewTargetNameInput
```

Field Value

Type	Description
InputField	

outputstring

String for the JSON ulpoaded to server when save gets clicked.

Declaration

```
public string outputstring
```

Field Value

Type	Description
System.String	

saveprevious

Save previous item to check if a new item was selected.

Declaration

```
public string saveprevious
```

Field Value

Type	Description
System.String	

SaveURL

Address for uploading the data when targets get updated in position etc..

Declaration

```
public const string SaveURL = "https://ar-app-rest-api.herokuapp.com/updateposition
s"
```

Field Value

Type	Description
System.String	

selected

Show if a new item gets selected by user to show location by changing color.

Declaration

```
public int selected
```

Field Value

Type	Description
System.Int32	

targetDataDropdown1

Dropdown menu filled with targest on first menu page.

Declaration

```
public TMP_Dropdown targetDataDropdown1
```

Field Value

Type	Description
TMP_Dropdown	

targetDataDropdown2

Dropdown menu filled with targets on last menu page.

Declaration

```
public TMP_Dropdown targetDataDropdown2
```

Field Value

Type	Description
TMP_Dropdown	

targetDataDropdown3

Dropdown menu filled with targets on adminpanel

Declaration

```
public TMP_Dropdown targetDataDropdown3
```

Field Value

Type	Description
TMP_Dropdown	

targetlist

List to store current targets.

Declaration

```
public List<Target> targetlist
```

Field Value

Type	Description
System.Collections.Generic.List<Target (DocFx.Target.html)>	

targetObjectPrefab

Prefab to create targets from.

Declaration

```
public GameObject targetObjectPrefab
```

Field Value

Type	Description
GameObject	

targetObjectsParentTransforms

Location were to store targets in environment.

Declaration

```
public Transform[] targetObjectsParentTransforms
```

Field Value

Type	Description
Transform[]	

targets

Array to store current targets.

Declaration

```
public Target[] targets
```

Field Value

Type	Description
Target (DocFx.Target.html)[]	

uniblue

Color of the blue used by the university.

Declaration

```
public Color32 uniblue
```

Field Value

Type	Description
Color32	

unired

Color of the red used by the university.

Declaration

```
public Color32 unired
```

Field Value

Type	Description
Color32	

value

Variable to store the selected value of a dropdown menu.

Declaration

```
public int value
```

Field Value

Type	Description
System.Int32	

Methods

addTarget()

Add a new target the the environment and update all dropdown menus.

Declaration

```
public void addTarget()
```

asyncGetRequest()

Method for getting the JSON Data from our WebServer.

Declaration

```
public async Task<string> asyncGetRequest()
```

Returns

Type	Description
System.Threading.Tasks.Task<System.String>	Returns the JSON as a Task string.

CreateTargetFacade(Target)

Creates the target in the environment with all given data from the parameter.

Declaration

```
public TargetFacade CreateTargetFacade(Target target)
```

Parameters

Type	Name	Description
Target (DocFx.Target.html)	*target*	Gets a target which gets created during method.

Returns

Type	Description
TargetFacade (DocFx.TargetFacade.html)	TargetFacade of each creates target.

FillDropdownWithTargetItems()

Fills both dropdowns with the names of the provided targets.

Declaration

```
public void FillDropdownWithTargetItems()
```

GenerateTargetItems()

Load the targets from the json to an array of targets and create targets in environment. Set the sliders to change the coordinates of the target selected. Update the Dropdownmenus with the names of the targets.

Declaration

```
public async void GenerateTargetItems()
```

GetCurrentlySelectedTarget(Int32)

Get the position data of currently selected target.

Declaration

```
public Vector3 GetCurrentlySelectedTarget(int selectedValue)
```

Parameters

Type	Name	Description
System.Int32	*selectedValue*	Value of dropdown to select a target.

Returns

Type	Description
Vector3	The Vector of the position data from the selectedValue provided.

makeallunselected()

If the user goes back to the menu all targets get unselected and their original color gets restored.

Declaration

```
public void makeallunselected()
```

removeTarget()

Remove a selected target from the environment.

Declaration

```
public void removeTarget()
```

restartpositions()

Deletes all changes made to the environment and loads the data from json again, in generel it reloads the last game settings.

Declaration

```
public void restartpositions()
```

SaveData()

Save all targets to the json to save the changes for the next users. Save the data to a string for uploading to the server.

Declaration

```
public void SaveData()
```

sendjson()

Save JSON to server and check if it worked.

Declaration

```
public IEnumerator sendjson()
```

Returns

Type	Description
System.Collections.IEnumerator	

SetSelectedChange2(Int32)

Notize when a target gets selected and set item like the value selected. Set the sliders to change the coordinates of the target selected.

Declaration

```
public void SetSelectedChange2(int selectedValue)
```

Parameters

Type	Name	Description
System.Int32	*selectedValue*	The value selected by the dropdown on third page.

SetSelectedChange3(Int32)

Notize when a target gets selected and set item like the value selected. Set the sliders to change the coordinates of the target selected.

Declaration

```
public void SetSelectedChange3(int selectedValue)
```

Parameters

Type	Name	Description
System.Int32	*selectedValue*	The value selected by the dropdown on third page.

SetSelectedTargetPositionWithDropdown(Int32)

Get the position of the target selected.

Declaration

```
public void SetSelectedTargetPositionWithDropdown(int selectedValue)
```

Parameters

Type	Name	Description
System.Int32	*selectedValue*	selectedValue by the dropdownmenu.

Start()

When the Game is started this method gets executed. It generates all targets, fills the dropdown menu with the targets and sets the paths were the json is stored.

Declaration

```
public void Start()
```

Update()

Method used to always check if the app is started. Checks if item is selected and always updates the position when slider gets moved. Changes color of the selected item from red to blue and backwards.

Declaration

```
public void Update()
```