Object by Object Documentation

Object-by-Object is a mobile gaming app for Android und iOS devices, which enables the discovery of exhibition objects outside of the museum's building, outdoors, thanks to augmented reality (AR). AR scenes are initiated by scanning a physical marker, which may be a QR code or an image. The virtual objects inside the AR scene are positioned in relation to the position of the physical marker.

The Object-by-Object app can be easily customised using the proprietary Content Management System Wezit Studio.

Gaming app Object by Object

App structure

The app is structured around several views managed by a *View Manager*, each of them containing a control script and several sub-components.

When a user navigates through the app, the suiting view will be enabled by the Manager

Splash screen and initialization

Screen

The splash screen can be customized in Wezit CMS, however it is important to know that it cannot be changed without updating the app on the store (contrary to other content that will update on startup). It consists of a single image that is loaded before compilation and is shown while the app data are loaded.

App initialization

When the app starts, it loads data in the following order:

- app config file (*config.json*) containing a link to the Wezit app manifest (which itself contains all needed link to access Wezit data)
- Wezit data:
    - Download the *manifest.json* file if it is missing or needs an update (which is checked using its remote and local ETag)
    - Download/update the content database file (*sqlite.sqlite*), settings file (*settings.json*), and assets file (*assets.json*) that respectively contain
        - The content tree and text content
        - The global app settings, e.g. app colour, home text, button text, …
        - Information on the assets used in the app, e.g. image uri, activity uri, …
    - Setup the different stores (see below) for easier data access later on
    - Download/update the assets used in the app, i.e. the images, videos, audios, 3D models, and activities
    - Initialize the player data
        - Create a *player.dat* file if it does not exist

        - Load the save data to save CPU usage by not having to load the text file each time
    - Initialize the global settings for easier global access, e.g. app colour, map pin sprites, challenge scores, …
- Start the first view (i.e. either language selection or home)

Stores

There are five stores that are used in the app:

- The tour store, granting quick access to every tour (museums in this context)
- The POI store, granting quick access to every POI (both tour POI and content)
- The POI location store, granting quick access to the geolocation of the POIs
- The cover store, granting access to video and music image covers

On top of that, the StoreAccessor references the current state of the app, which is composed of the following elements:

- Current *KioskState*, i.e. current view
- Current language used by the app
- Access to the afore-mentioned stores
- Current selected POI
- Current selected tour
- Current selected image bank of the tour

Language selection

The language selection happens once upon first use of the app. Later on, it is possible to modify the selected language via the app's main menu.

If only one language is available, this screen is not shown. The corresponding menu link will have to be disabled on the Wezit CMS beforehand.

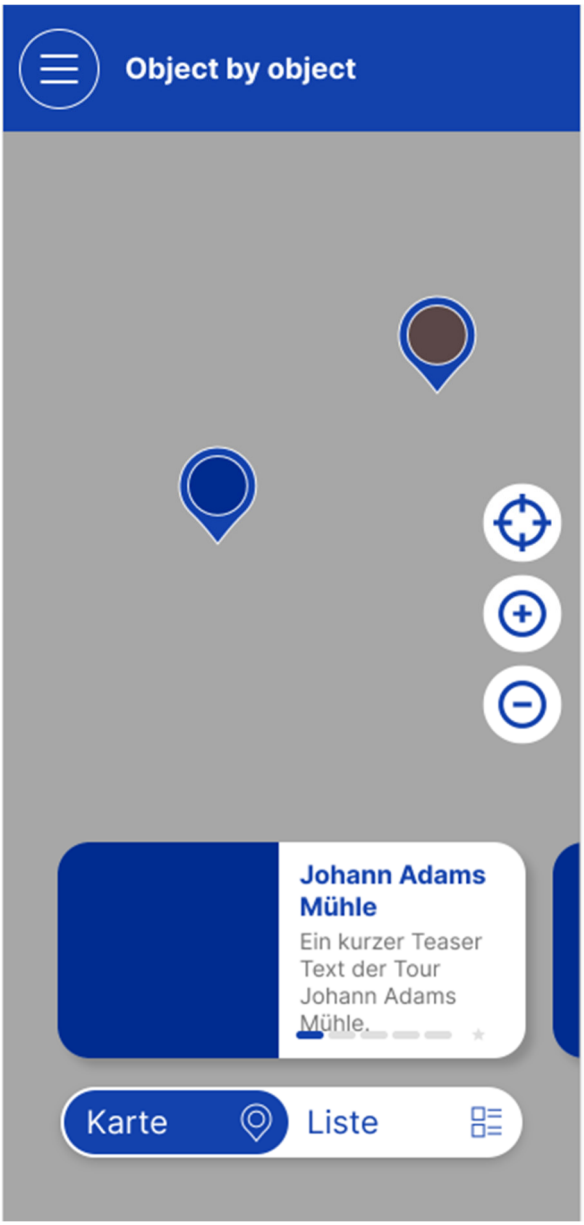
Introduction

The introduction is shown once upon first use of the app. From the introduction users can access the tour selection page.

Tour selection

All available tours are shown. Users can select their tour based on the information provided in the teasers for each tour, which lead to a description of the tour, and on their proximity to the tour location as shown on the OpenStreetMap map of all tours.

The map is managed using the *OnlineMap* plugin and instantiating markers at the position written in the Wezit CMS data.



Playing mode

After choosing a tour users select their playing mode. There are two playing modes. In the Discovery Mode, users freely explore the tour's points of interest (POI). In the Challenge-Mode users earn points and must answer quiz questions for each POI to earn bonus points.

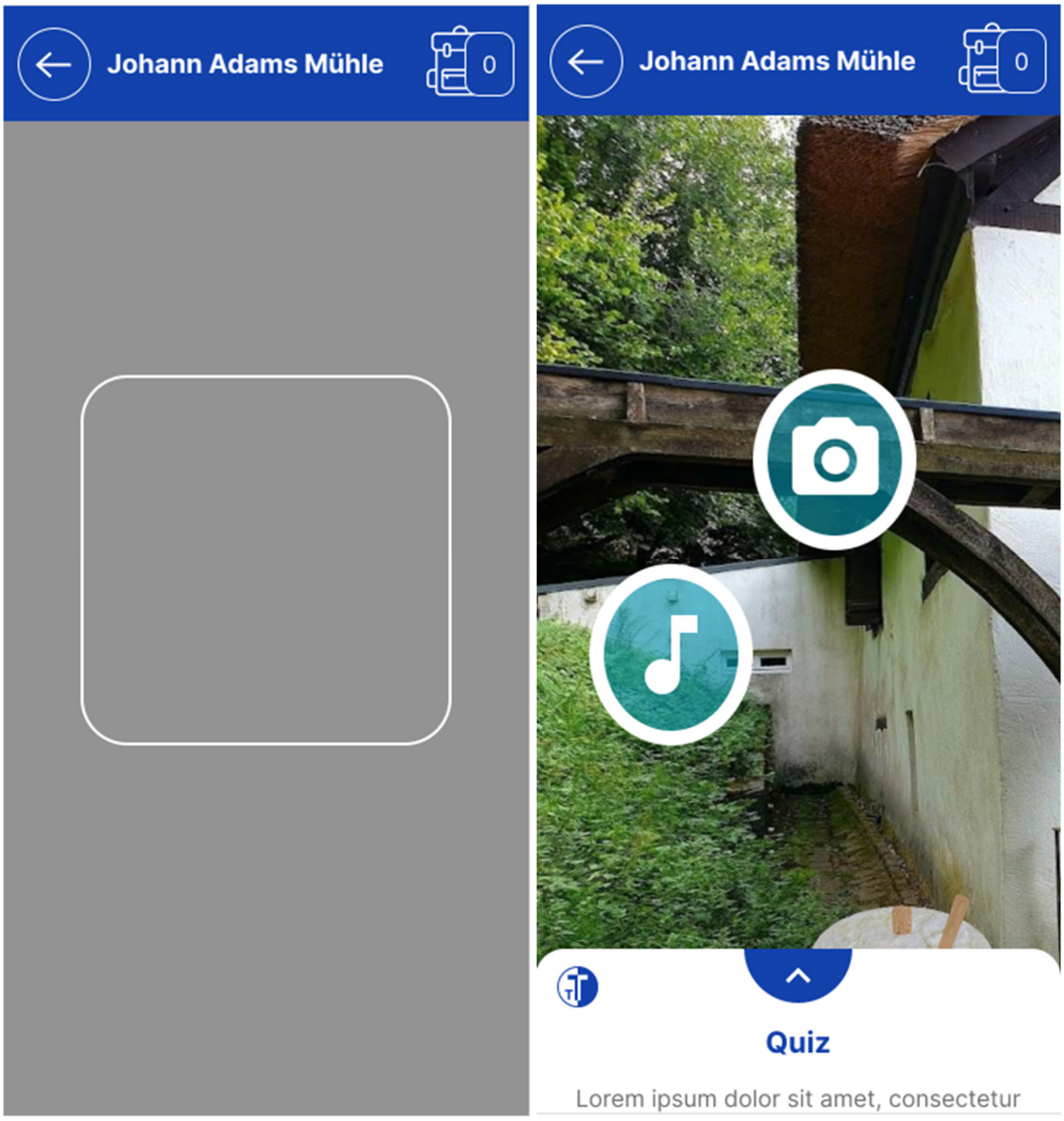
Tour map (Tour start)

After choosing a playing mode, users access a map of the tour showing the position of the different POI.

AR Session

Using the tour map, users go to their first chosen POI and look for the corresponding AR marker. By scanning the AR marker, they start the first AR session.

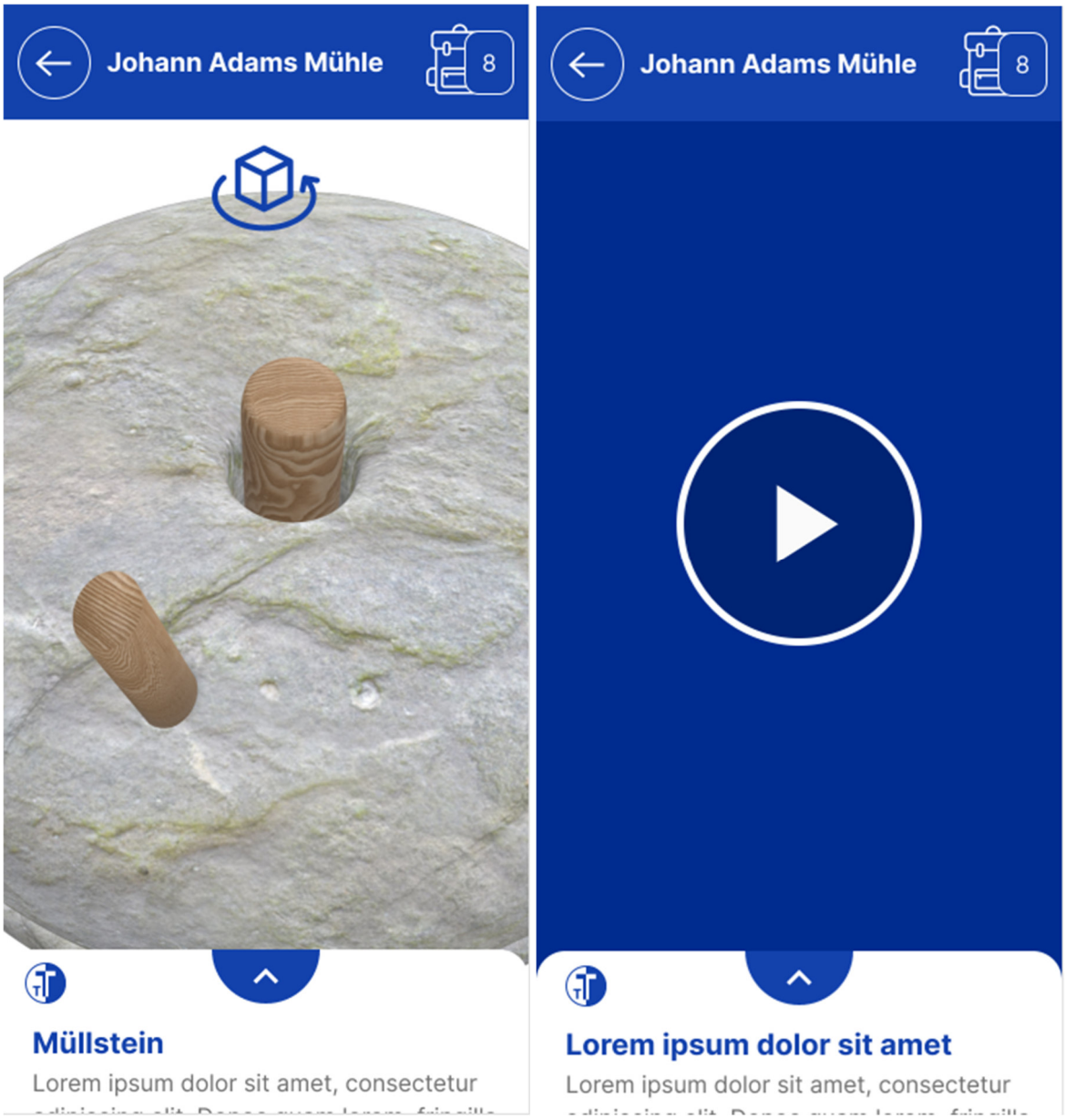
The image recognition is achieved with Unity AR Foundation's *AR tracked image manager*. When initializing the view content, we create a new runtime reference library containing the POI reference image set in the Wezit CMS. AR Foundation will then look for this image and inform the script when it is detected, along with its position. It will then use real-world reference points



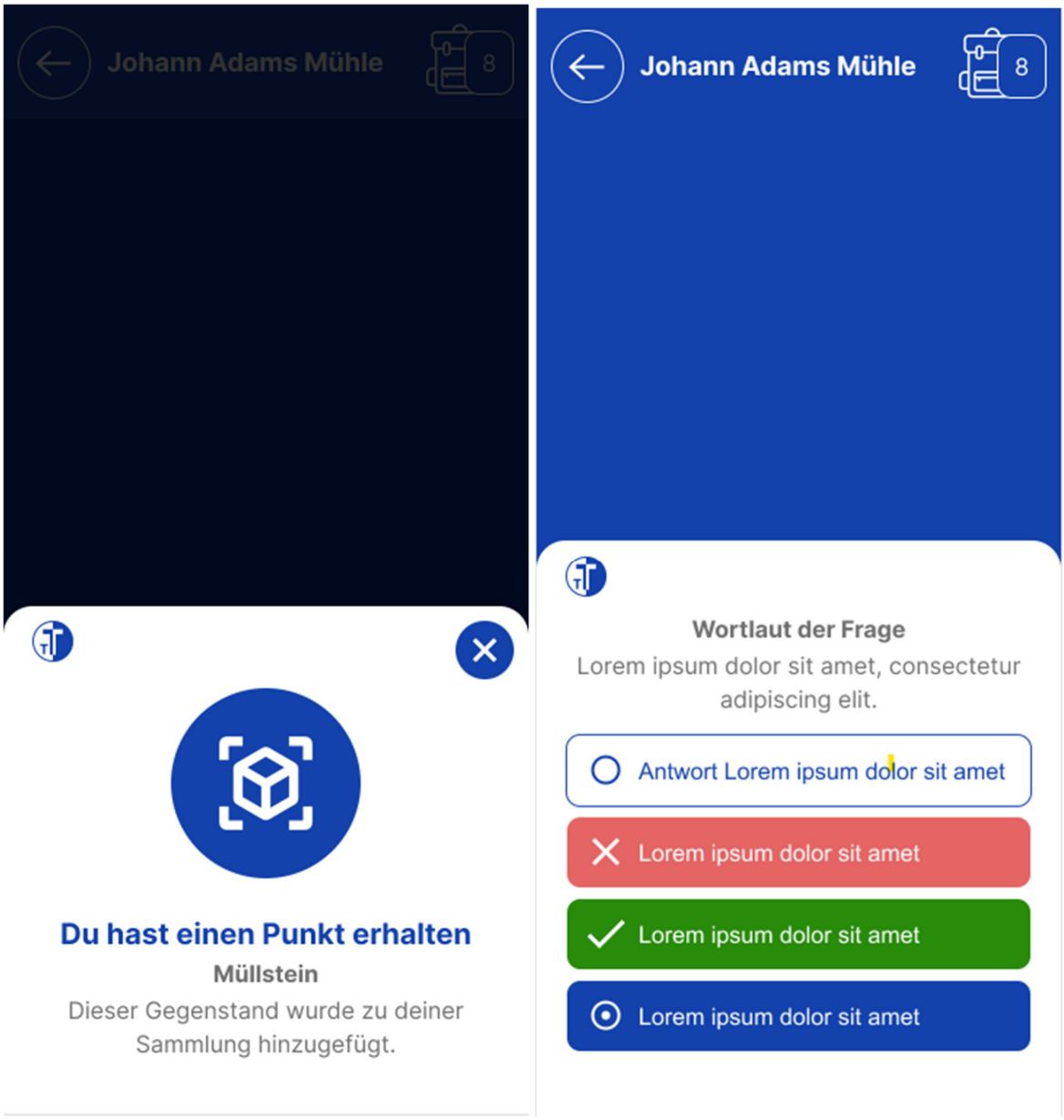
Objects

There are 4 types of virtual objects in any AR session:

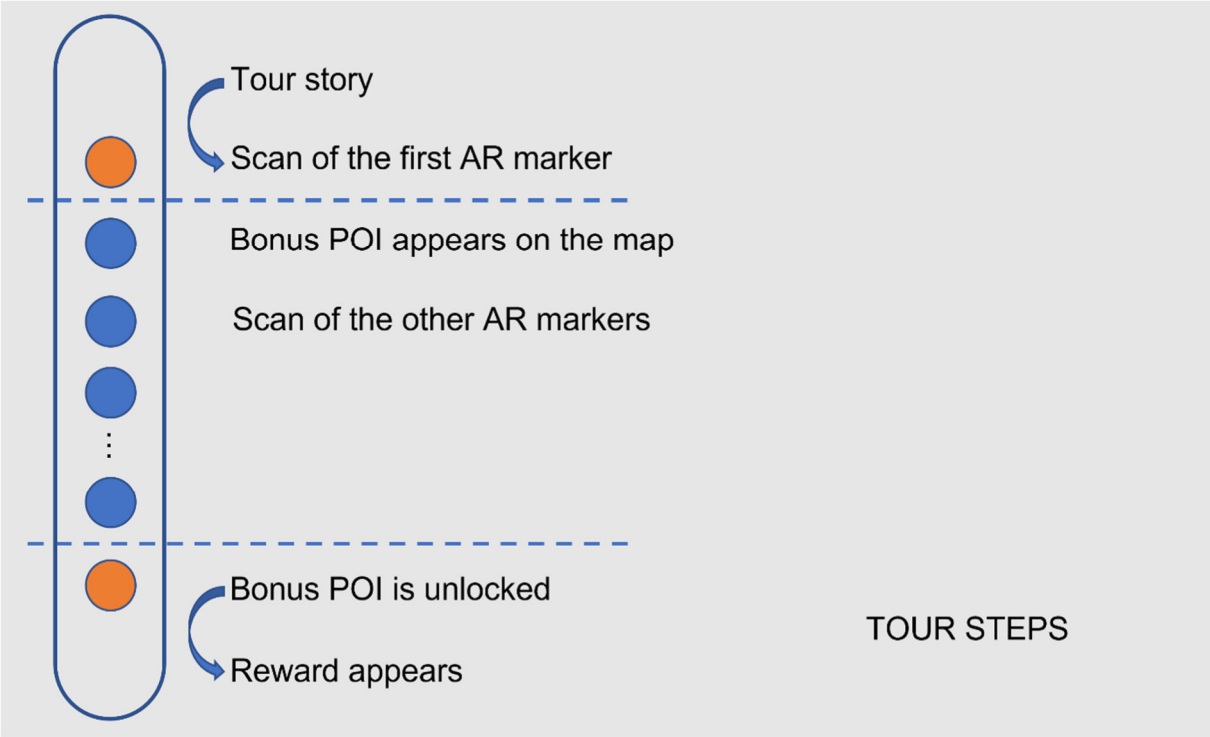
- 3D-model of an exhibition object
- Video
- Audio
- Image



In the Challenge Mode users earn one point for each virtual object they interact with. In addition, users are presented with a quiz question they can answer at any time while they are at the POI location.



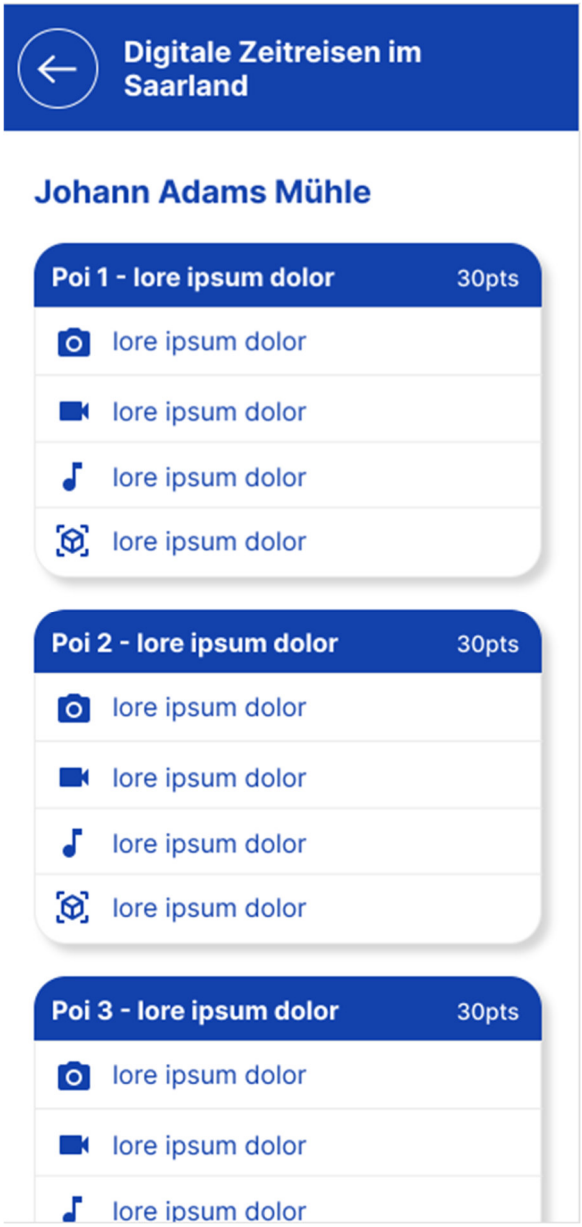
Tour process



Once users visited their first POI, a bonus POI appears on the tour map. This bonus POI gives access to the end of the tour story. If users decide to open the bonus POI without having visited all POI on the tour, they are not eligible for the tour reward. If users have visited all POI and then open the bonus POI, they receive the tour reward in the form of a scratch and reveal mini game.

Collection

All virtual objects that users interact with during the course of the tour are added to their collection. Objects in the collection can be accessed at any time, even when users are no longer on location. Users playing in Challenge Mode also see the points they have earned in their collection.



Gaming app installation and set-up

Download

The repository containing the Unity project can be found here. This allows compilation for both Android (resulting in a .apk file that can be installed on Android phones) and iOS (resulting in a Xcode project that must be compiled on Xcode).

Access token

No access token is needed to connect to the Wezit CMS, however you must change the *manifestUrl* field of the *config.json* file (*project_root/Assets/StreamingAssets/config.json*, line 49) to the one used by the corresponding Wezit content application before compilation.

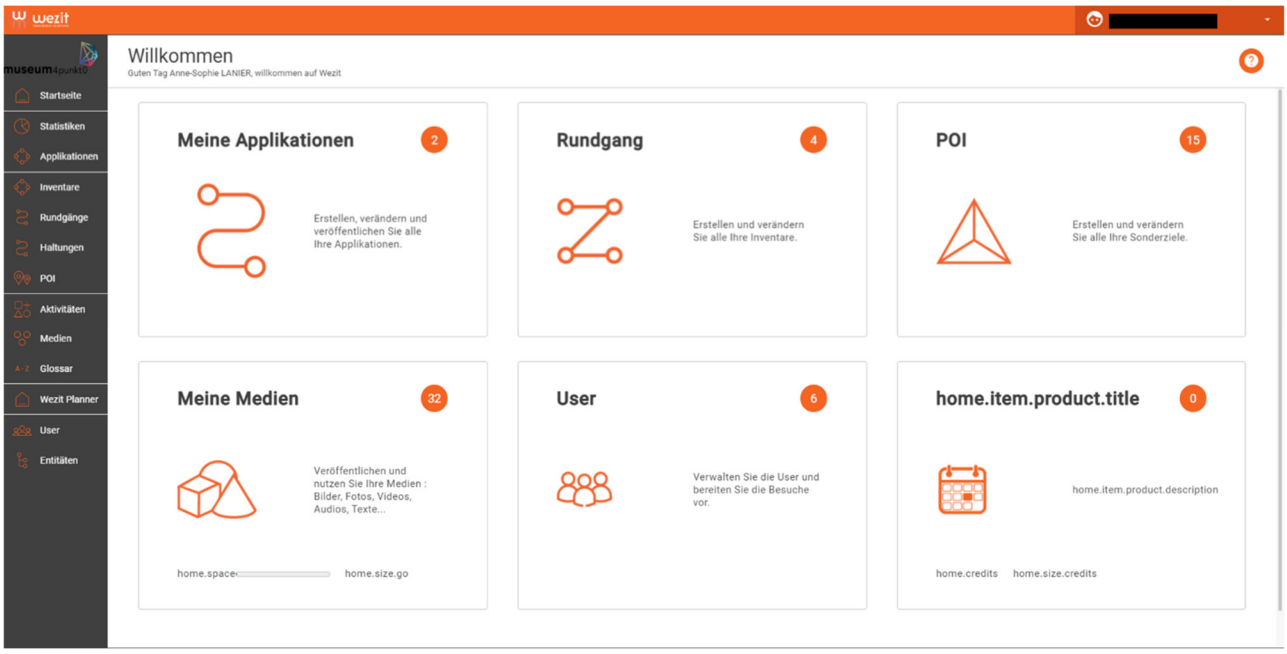
Content Management System

Basic structure

The gaming app Object by Object can be set up and customized easily with the proprietary Content Management System (CMS) Wezit Studio.

Tours, POI, Media

Wezit Studio is structured around tours, points of interest (POI) and media assets.



Tours can be accessed and created under the Tours menu item. The POI associated with the tour are listed there but are also accessible under the POI menu item, which gives access to all POI for all tours. The media assets are accessible both via the Media menu item as well as via the POI with which they are linked.

Positioning of objects in Augmented Reality

The image or QR code used on the physical AR marker is saved as a media asset to the POI in Wezit Studio. The actual position of the physical marker is used as the reference point for the positioning of the virtual objects in the AR scene.

To simplify positioning (see following paragraph), the angle to the ground (in degrees) and the size of the physical marker (in meters) are written in the *space* and the *extent* fields respectively.

Two fields in the metadata for each object allow to position the object in relation to the physical marker: the *localisation* field (0,0,0) allowing to provide the distance of the object to the marker along the x, y and z axis, and the *space* field (0,0,0) allowing to specify the angle of the object to the physical marker.

Customisation of the gaming app in Wezit

The simplest way to customise the look & feel as well as the contents of the gaming app is via Wezit Studio. The application settings can be adapted to modify the main colour of the app, icons and images, the text of the navigation elements as well as the structure of the app menu. Content (POI and objects) can be added as POI with corresponding custom media and grouped into tours.

Structure of the content

The content must follow the following structure

- Application: this is where global settings will be modified, all tours must be put as children
    - Tour: the museum exhibition, must be geolocated and contain two images. The reference picture will be used by the map pin
        - POI: the geolocated item the contains all AR objects. It must contain two images. The reference picture will be used as the AR marker.
          The *localization* field indicates the angle relative to the ground, while the *extent* field indicates the size of the marker (optional)
            - Content child: the audio, video, image, or 3D content. It must contain a direct relation to the content that will be displayed,

and be typed accordingly using the *Type* field. It can be positioned relative to the QR code as mentioned before

- Secret POI: The “secret POI” is the bonus POI containing the scratch activity won after scanning all POIs. It contains two images, of which the reference picture will be used in the selfie view. Its *Type* field must be “secret”

- Image bank: The image bank contains the images used in the different popins. Its *Type* field must be “bank”. It is optional, and no popin icon will be used for the tour if it is missing.
    - QRCode popin icon: *Type* field “QRCode”
    - Default popin icon: *Type* field “main”
    - Negative popin icon: *Type* field “bad”
    - Positive popin icon: *Type* field “good”

CMS access

To inquire about a Wezit Studio licence and receive access, please send an e-mail to info@wezit.de.