**Notes**

*This file contains notes, issues, ideas that can be implemented later, and pros and cons about using different features*

**Museums**

-At the museum, we can’t use the “Scale” button, because each model must have a parent that dictates the size of the objects. Maybe we can find a method the get the size of objects, that do not have one parent

**Code stripping**

-If managed code stripping is set to high, the app freezes at loading

**Flashlight**

-We couldn’t use the device’s flashlight. I tried methods from Vuforia, JavaClass, MyAwesomeFlashlight from github, but none of them worked. We asked it on stack overflow, and it looks like, it can only be used if the app doesn’t use ARCore, but ARCore boosts the performance of the tracking, so it shouldn’t be disabled

**Videoplayer**

-If the video is saved at the files, it can’t be deleted, because when it is played, the video is read from the path, the video should be saved across all the usage of the museum and only deleted after, which is not possible because, it’s not possible to delete something when the application is closed, at least on android. If the video is read from url, and the VideoPlayer.Stop() is called (it is always called, when the GameObject, to which the VideoPlayer is attached is deactivated), the app crashed on android. The url option is better, but it requires internet connection

**Security**

-Directory listing must be disabled on the settings of the server, because if someone gets the url of the server, that person can fin all the files that are on the server

-It’s possible to make a server with back-end, to completely secure the files, but it’s complicated

-The videos and sounds aren’t encrypted, we couldn’t read the encrypted files from path

**Games**

It is probably better to keep the games in different scenes, because if they are in the Main scene, they consume memory, even if the GameObjects are innactive and the scene itself doesn't take up much memory. This is not the case for the museums and the modelplacement, because they use many shared components, so they should be in the same scene