AHS-AR ADMINISTRATOR MANUAL

Install AHS-AR for development, program overview and class diagram, licensing, and troubleshooting

AHS-AR
Administrator Manual

Runtime Setup

Trainer PC

1. Visit our repository at: https://github.com/401AR/W17401AR
2. Click the green “Clone or Download” button, then select either clone, or download as zip. If you are not using GitHub, select download as zip.
    a. If downloading as zip, extract the archive to your intended working directory. Do not extract to the Program Files directories, as read/write restrictions are in place in those folders.
3. Run the executable at \W17401AR\AHS-AR.exe

Trainee Android

1. If the repository has not been downloaded yet, follow steps 1 and 2 on the Trainer PC
2. Turn on the android device and plug them into the Trainer PC via usb.
3. Transfer the \W17401AR\ AHS-AR.apk file to the download folder in the android device.
4. From the android device, open the file AHS-AR.apk file, and install the app.
5. Return to the home screen, navigate to apps, and run the installed application

Development Setup

To further extend AHS-AR, a full installation of the development environment is necessary.

Prerequisites

All prerequisites must be installed in order to open and build our GitHub repository properly.

Unity

AHS-AR was developed using Unity 5.5.0f3 (64 bit). In order to minimize conflicts, we highly recommend you use the same version and architecture.

- Download and install Unity 5.5.0f3 (64 bit) from: https://unity3d.com/unity/whats-new/unity-5.5.0
- During installation, make sure to select all Visual Studio (see IDE below), C# , Unity, and Android options.

Please keep in mind that the install of Visual Studio can take several hours.

IDE (C#)

An IDE is required to properly write and debug code changes to AHS-AR. Visual Studio (VS) Community edition is recommended, as it will cause the least complications with the development environment.

- Either MonoDevelop or Visual Studio Community edition can be installed while installing Unity.

VS Community is recommended, but is a heavy installation (~ 8GB). Depending on your development needs, purchasing a Visual Studio license may be necessary.

Android 4.0.4 SDK

This is installed along with Visual Studio. If you opted not to install Visual Studio, then you will need to download and install Android Studio (and SDK manager).

Note: If using the Epson BT200 glasses, Any Android version above 4.0.4 is **not** compatible with EPSON BT200.

Vuforia Unity Extension 5.3.6 (QR Handling)

The Unity package for Vuforia is included in the AHSAR project already. It does not need to be downloaded and imported again. The included version of the package is no longer available for download.

SDK Documentation

https://library.vuforia.com/all-articles

SQLite Database Tool

SQLite Browser is a lightweight SQLite tool that can read and save the .bytes extension. This extension is required for the SimpleSQL plugin used by AHS-AR.
http://sqlitebrowser.org/

Adherence to the license of SQLite Browser is only necessary if modifying or redistributing SQLite Browser itself.

3D Modelling Program (Optional)

Blender, 3DSMax, Unity, or other similar modelling programs can be installed and used to modify the provided baby models.

Product Installation

4. Visit the repository at: https://github.com/401AR/W17401AR
5. Click the green “Clone or Download” button, then select either clone, or download as zip. If you are not using GitHub, select download as zip.
    a. If downloading as zip, extract the archive to your intended working directory. Do not extract to the Program Files directories, as read/write restrictions are in place in those folders.
6. Open Unity, select “Open Project” and then select the project file within the archive or git clone. Alternatively, double click W17401AR\Assets\Scenes\Offline.scene.
7. After installing all prerequisites, build and run the project. If everything has been set up correctly, the build will succeed.

Build Instructions

All product development is done in C#.

The Trainer executable:

- Must be built for PC.

The Trainee executable has additional requirements:

- EPSON AR Glasses (BT200) must be built for Android mobile, targeting Android 4.0.4.
- Any Android version above 4.0.4 is **not** compatible with EPSON BT200.

Network Setup

AHS-AR operates over a local area network wireless connection. A successful network connection between the Trainer and Trainee applications, requires:

1. Trainer and Trainee devices must be connected to the same wireless network.
2. Trainer and Trainee must have active connections.
3. Trainer and Trainee must allow AHS-AR.exe unrestricted access over both TCP and UDP through port 7777. Opening this port to other programs is unnecessary.
4. The config.txt file must specify the correct local IP (not public IP) of the Trainer’s computer, and must specify port 7777.

Changing the Server IP

The Server local IP (not the public IP) must be specified in *config.txt*. *config.txt* can be found under the asset root directory.
The default contents of config.txt are:
; This is an ini comment
[Server]
IP=172.28.102.232
Port=7777

Despite the *.txt* extension, *config.txt* is read and written to as an ini file. Any changes will need to follow the ini format originally provided.

This config file must be changed on the Trainer PC and on the Trainee Android unit attached to the AR Glasses.

Licensing

Unity

If your company (or incorporated entity) made over $100,000 USD in the last fiscal year, you must purchase a Unity license to sell AHS-AR.
http://download.unity3d.com/unity/licenses

Visual Studio IDE

A full breakdown of products is available here:
https://www.visualstudio.com/vs/compare/

To generalize, usage of the Community Edition does not violate licensing unless you are running an enterprise. An enterprise is classified by Microsoft as an organization with more than 250 computers or greater than $1,000,000 USD in revenue.

Unity Plugins

AHS-AR makes use of a few additional plugins to maximize cross-compatibility.

SimpleSQL

SimpleSQL was purchased for cross-platform database handling. Since the plugin is already included in AHS-AR no further complications are expected.

Asset Store: https://www.assetstore.unity3d.com/en/#!/content/3845
A support link is included in the Asset Store page.

IniParser

IniParser is a free cross-platform ini handler. Since the plugin is already included in AHS-AR no further complications are expected.

Asset Store: https://www.assetstore.unity3d.com/en/#!/content/23706
A support link is included in the Asset Store page.

AHS-AR

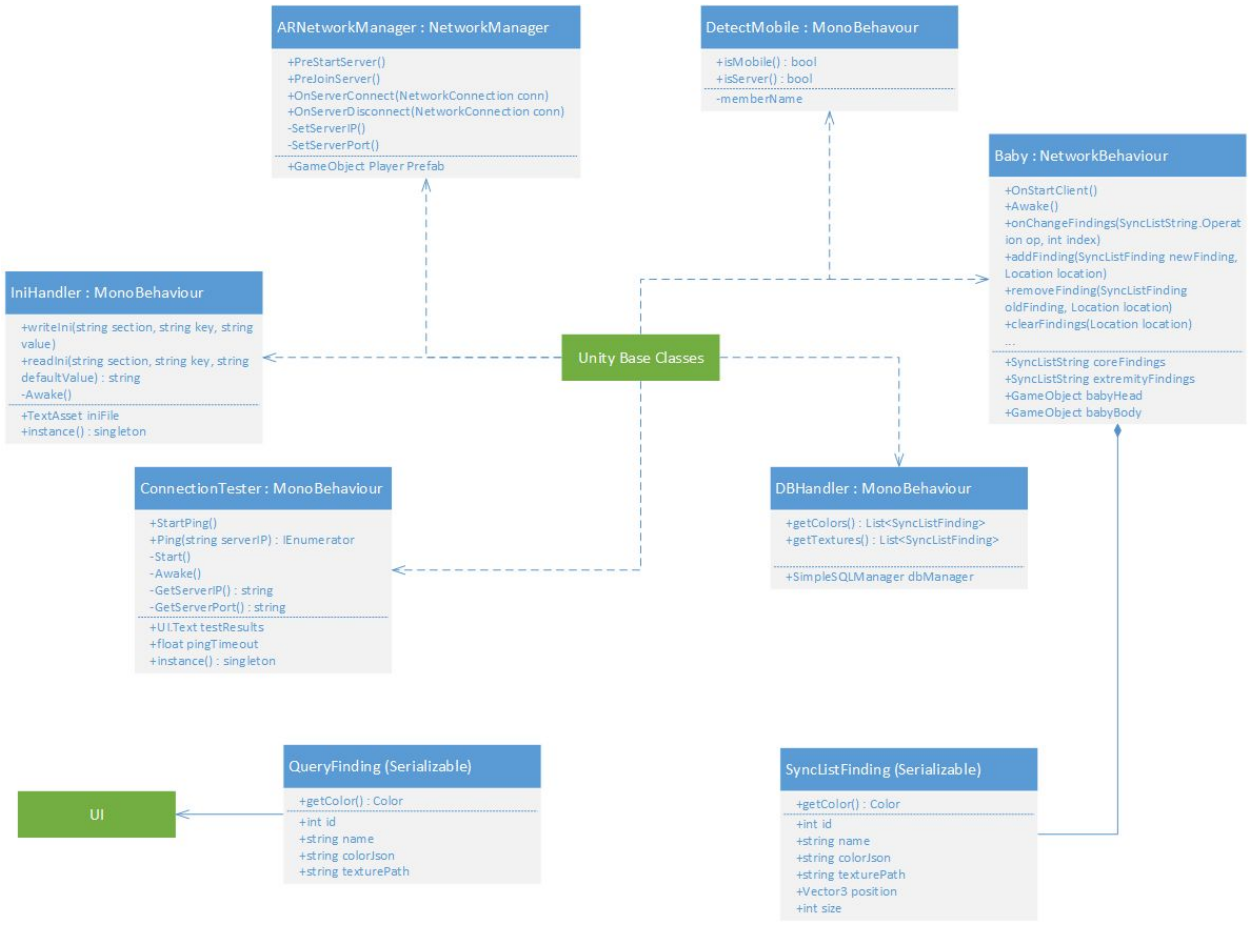
Overview

AHS-AR is split into two scenes; Multiplayer.scene and Offline.scene. Offline.scene is meant to be the default scene, as it is responsible for establishing network connections and maintaining an instance of Network Manager. Connection failures redirect back to Offline.scene. Server IP and port are specified in config.txt, so changing server IPs are not specified at run time and do not require rebuilds..

The training simulation itself takes place in Multiplayer.scene. Trainer and Trainee both participate in Multiplayer.scene. Network Identities are used to mark which aspects of the scene and UI will be visible to each. The Trainer UI is populated by all findings from findings.bytes (the findings database). Changes applied by the Trainer (acting as the server) are pushed to the Trainee (acting as the client). The Trainee is blocked from making any changes to the baby data object. In order to make use of network synchronization, all synchronized objects are in JSON format.

When the Trainee joins the server, the trainee loads into Multiplayer.scene where a AR camera is loaded instead of the UI scene the trainer sees. The trainer will see nothing on screen (besides a vuforia logo), but when the trainee looks at the target baby, they will see an overlay of the baby appear over the baby. All changes made to the baby will be updated and seen by the trainee as long as they are connected. If the trainee gets disconnected, they will not get any changes made before they reconnect.

Class Diagram (Core)



Additional Limitations

The SimpleSQL plugin causes a few additional limitations:

- The findings database must be housed on the Trainer PC. It cannot be housed on a separate server. In order to host the database on a remote server, a different SQL solution will be necessary.
- Windows Phone (WP8) and Windows Store are not compatible with this plugin. This is because Microsoft is retiring WP8.
- Database extension must be .bytes. This is for cross-platform compatibility.

Troubleshooting

Development Issues

Black Screen in Multiplayer.scene on a Computer

Attempting to test the Multiplayer scene directly will result in a black screen. Testing needs to start with the Offline scene. Once a connection is established, the NetworkManager will transition to the Multiplayer scene.

The Multiplayer scene requires connection information that can only be provided through the Offline scene.

Database Won't Load

Check that when building the Trainer application (for Windows/Linux/iOS), the build platform selected is architecture "*x86_64*" (64 bit architecture) and not "*x86*." The AHS-AR Unity application is designed for 64 bit architectures, and will not find the SQLite libraries when built in "*x86*."

If the database is specified correctly and still will not load, please follow the procedure outlined below:
1. Open up the existing database with DB Browser for SQLite
2. Export -> Save as SQL
3. In the Unity project, multiplayer scene, check that you can't drag-n-drop onto SQLManager (under DBManager).
a. If not, the connection between SimpleSQLManager and the database file has been corrupted.

All further steps assume this link is corrupted.

4. In the top Unity Menu bar, select Tools -> Create New Database. This database appears in the root Assets folder
5. Delete the old database
6. Assign the same name to the new database as the old database.
7. Now check that you can drag-n-drop the new database into the SimpleSQLManager component.
a. If not, the new database is corrupted. Delete it and retry steps 4 - 7.
8. Now, open up the new database with DB Browser for SQLite.

9. Using DB Browser... Import -> SQL. When prompted to open a new database for changes, select No.
10. Confirm that the new database has all expected contents, then save.
11. Now check that you can drag-n-drop the new database into the SimpleSQLManager component.
a. If not, then DB Browser made a bad write to the file, and has corrupted the database. Delete it, and retry steps 4 - 11.
12. Now you've confirmed the database is content-restored and the connection between it and SQL Manager is valid.
13. Now move the database into Assets/StreamingAssets. Otherwise, Unity will not include the database resource in builds.
14. Now check one last time that you can drag-n-drop the new database (in StreamingAssets) into the SimpleSQLManager component. SQLManager is finicky about the DB being in the StreamingAssets folder.
a. If not, the connection between SimpleSQLManager and the database got corrupted, and you'll need to retry steps 4 - 14.
15. Make sure that the "overwrite" and "change working copy" checkboxes of SQLManager are unchecked. If they aren't, uncheck them. This is to avoid potential confusion over the database path in the built product.
16. Build and test.
17. Now check one last time that you can drag-n-drop the new database (in StreamingAssets) into the SimpleSQLManager component.
a. If not, then, you guessed it, redo steps 4 - 17.
18. If you survived this far then the database and SimpleSQLManager have valid references and are working properly.

Contact Us

If the common issues here do not help, you are welcome to contact us:

- **Avery Tan:** avery_tan@hotmail.com
- **Christopher Hegberg:** hegberg@ualberta.ca
- **Gemma Marcinkoski:** gemmamarcinkoski@gmail.com
- **Laura Patricia Juarez Reyes:** ljuarezr@ualberta.ca
- **Michael Morris:** mcmorris@ualberta.ca