# ARLearn: augmented reality meets augmented virtuality



**DOI:**
10.3217/jucs-018-15-2143

**Document status and date:**
Published: 01/08/2012

**Document Version:**
Peer reviewed version

# ARLearn: augmented reality meets augmented virtuality

**Stefaan Ternier**
(Open University of the Netherlands, The Netherlands,
stefaan.ternier@ou.nl)

**Roland Klemke**
(Open University of the Netherlands, The Netherlands,
roland.klemke@ou.nl)

**Marco Kalz**
(Open University of the Netherlands, The Netherlands,
marco.kalz@ou.nl)

**Patricia van Ulzen**
(Open University of the Netherlands, The Netherlands,
patricia.vanulzen@ou.nl)

**Marcus Specht**
(Open University of the Netherlands, The Netherlands,
marcus.specht@ou.nl)

**Abstract:** This article deals with educational opportunities for mixed reality games and related scenarios for learning. It discusses several issues and educational challenges to be tackled when linking augmented reality and augmented virtuality. Second, the paper describes the architecture of the ARLearn system which offers highly flexible support for different educational settings. Three prototypical use cases implemented based on the underlying ARLearn framework are discussed, which are a field trip system, an augmented Google StreetView client called StreetLearn, and a real time crisis intervention game. ARLearn combines real time notification and mixed reality games across Mobile Augmented Reality and Virtual Reality and the authors aim to use the underlying (open source) framework for further case studies and mixed reality applications for learning support.

**Keywords:** immersive learning games, augmented reality, augmented virtuality
**Categories:** L.5.1

## 1 Introduction

Recently augmented reality has gained a lot of commercial interest and research interest also due to the fact that mobile devices have gained the possibility to support situated mobile augmented reality applications. Beside the classical approaches of high tech lab augmentations this opened AR technology to a huge user group of smartphone owners. As described in [Specht, Ternier & Greller, 11] mobile augmented reality applications for education implement learning support and linking of the real world context of a user and the digital overlay according to different patterns. In our previous works we applied mobile augmented reality to field trip support, and augmented reality games [Specht, Ternier & Greller, 11].

Additionally there is a rising market of educational games in general [PWC, 10] and especially for networked games including social aspects as MMORPGs (Massively Multiplayer Online Role-Playing Games). When talking about educational applications of these technologies several problems are discussed such as the transfer of experience made in virtual worlds to their real world application. In recent times immersive games are discussed to overcome problems like the separation of real world problems and educational contexts, the transfer of knowledge from one application context to a different one or the general reduction of inert knowledge. In our view the combination of features of mobile augmented reality, and educational simulations, microworlds, and online games provide a unique opportunity for linking game experiences and real world application contexts.

In this article we introduce the ARLearn system - an architecture that can cover a wide range of educational designs, immersion levels, and game logics. In the first section we discuss related work and define our challenges and hypotheses. In the second section, the relation between augmented reality and augmented virtuality is discussed. Next we introduce the architecture of the ARLearn environment and its main concepts. In the third part three case studies are introduced and discussed. Finally, we discuss our findings and present our plans for future work.

## 2 Educational Background

Several educational theories are related to the goal of embedding learning processes into real world application and performance. The anchored instruction approach [Bransford et al., 90] was developed to decrease the problem of inert knowledge through the presentation of real authentic problems and the active exploration by learners. Augmented reality (AR) and AR games offer a unique opportunity to implement this core idea in linking real world situations and problems with learning support. The theory of situated learning [Lave & Wenger, 90] is grounded on the assumption that learners do not learn via the plain acquisition of knowledge but they learn via the active participation in frameworks and social contexts with a specific social engagement structure. Kolb's learning cycle [Kolb, 84] and the concept of experiential learning discusses four stages for the process of (adult) learning:

1. Concrete experience
2. Reflection
3. Abstract conceptualisation
4. Active experimentation

[Dede, 09] defines immersive learning as learning that involves the "subjective impression that one is participating in a comprehensive, realistic experience". In her review of immersive games, de Freitas stresses the importance of linking the experiences made in a game, simulation or micro world with their application in real world practices [de Freitas, 06]. [Brown & Cairns, 04] describe game immersion as a continuum from engagement over engrossment to total immersion. [Dede, 09] differentiates between the following types of immersion:

- *Actional immersion* enables an individual to have experiences which would be impossible in the real world.
- *Symbolic immersion* involves the triggering of semantic and psychological associations via the content presented.
- *Sensory immersion* replicates the experience of a remote location via haptic feedback.

Immersive games can offer an authentic learning context. They can help to mimic a social engagement structure and can potentially cover all phases of the experiential learning cycle by Kolb [Kolb, 84]. But it is questionable, especially from an economical perspective, if it makes sense to model all parts of the cycle as realistic components of a game. [Herrington, Reeves & Oliver, 07] discuss the question for which purpose it makes sense to fully mimic a situation from real-life into a virtual representation and they ask the critical question how "real" an immersive learning environment needs to be to support a specific learning experience. The authors point to the issue that maximum fidelity does not necessarily lead to a maximum of effects on the learning outcomes. [Dede, 09] discusses the three promising aspects of immersive learning:

1. *Multiple perspectives*: To offer a switch between an egocentric perspective and an exocentric perspective can be a powerful means to lead learners to innovative problem solutions. According to the author the egocentric perspectives supports the actional immersion and motivation through embodied, concrete learning, while the exocentric perspectives supports more abstract, symbolic insights gained from positioning oneself outside the concrete context.
2. *Authentic Problems*: Since a situated learning approach is hard to realize in a traditional classroom setting immersive interfaces or situated simulations [Liestøl, 11] offer a promising way to create authentic problem finding and problem solving communities in which learners interact with other actors (computer generated or "real"). Last but not least the author discusses the importance of far transfer.
3. *Transfer*: Dede discusses the problem that *far transfer*, being the application of knowledge in a completely different context than the one where the knowledge and skills were achieved, is one of the biggest problems of educational systems worldwide. Immersive interfaces offer a "design" context in which situations can be developed and tested to further research educational design for far transfer.

Despite the huge potential of immersive games to overcome the gap between the real world and the educational context and the rising market for electronic games [PWC, 10], the use of technology-enhanced immersive games in education is still quite low. The reasons for this are manyfold:

- high game development costs meet limited educational budgets [Westera et al., 08]
- predefined games are hard to be integrated in the educational process [Klopfer, Osterweil & Salen, 09]
- learner support in online games does not easily scale [Van Rosmalen et al., 08]
- furthermore, game platforms up to now could not easily be integrated with real world environments.

Considering the above described benefit and problems, the authors have developed a framework called ARLearn [ARLearn, 2011]. This framework enables us to define an underlying instructional design for games and has linked different user clients as a mobile augmented reality application and a virtual world to support real time communication, notification and updating in both directions. In the following we will introduce the technological background of the framework and its implementation.

## 3 Linking Mobile Augmented Reality to Augmented Virtuality

Milgram's Virtuality Continuum [Milgram, 94] presents an axis where mixed reality extends from real environments to completely virtual environments. Augmented reality and augmented virtuality applications are ranging somewhere on this axis. In applications for Augmented Virtuality (AV) the concept of immersion is important. Navigating through a synthetic world, a participant can interact with either fictional or real objects. Applications for augmented reality (AR) build upon a real environment. AR applications add virtual media to a real environment, presenting information that is not visible in the real world.


Figure 1: Milgram's Virtuality Continuum

With the availability of technologies for virtual navigation the situation to design immersive games has changed. Virtual Navigation solutions as Google StreetView enable new possibilities to use existing models of real world environments without the additional effort to model every detail of the targeted educational environment.

Furthermore state-of-the-art augmented reality browsers like Layar and Wikitude enable connecting virtual elements to a real world environment and enable the user to filter information according to the current context. Recently, the authors have implemented first mashups for Google StreetView (called StreetLearn) and for mobile devices which use the Android Google Maps API (called ARLearn). StreetLearn is intended to provide an augmented virtuality environment on a Desktop, while mobile devices are provided with an augmented reality experience through ARLearn. By creating scripts, adding interactive elements and by introducing gamification elements, we believe that we can increase the learner's motivation and provide a richer learning experience linking mobile augmented reality and augmented virtuality.

Several examples from literature follow comparable approaches and propose instructional design approaches for immersive games linking AR and AV. [Appelman, 05] argues for combined teams of instructional designers and game designers to work together and presents an approach for immersive game design fostering this collaboration. [Doswell and Harmeyer, 07] combine research in AR with a mobile learning game including a virtual tutoring approach. However, their approach requires specific hardware to be used (mobile head-up displays in wearable glasses) and relies on a complex system architecture. Also, the games realised for their platform require a high modelling effort and specialist involvement in game creation. [Santamarina et al., 10] paid attention to the authoring aspect for mobile learning games by providing an authoring environment directed to instructors. To author a new game, instructors can combine pre-defined mini-games to larger game scenarios. A general shortcoming of this approach is, that the author can only choose among the existing mini-game types, while the creation of these mini-games again requires technological expert knowledge. Additionally, their approach relies on portable game-consoles rather than on all purpose devices.

Building on these results, the authors aim to provide an approach that relies on freely available technology distributed via general purpose devices (mobile phones and web-browsers on stationary computers). Furthermore, the authors aim for an architecture, that allows serious games to be created with no technological expertise. In this sense the ARLearn architecture fits very well in the classification contributed by Milgram and Kishino as the architecture supports games that are played both in a real and virtual environment [Milgram and Kishino, 94].

Two core motivations led to the development of the ARLearn architecture:

1. Game environments, which enable game plays that are synchronised between augmented reality and augmented virtuality are sparse, especially, when looking at an educational context. Even though game patterns and game designs that can be used in augmented reality and augmented virtuality are basically the same, mobile games in augmented reality are up to now developed independently of games for augmented virtuality environments.

   *Research question (rq1):* Can we develop a common architecture, that allows to define, develop, and play location-based and context-aware learning games ready for delivery in augmented reality using mobile devices as well in augmented virtuality using stationary computers, while still meeting simplicity and flexibility requirements?

2. Creating learning games for augmented reality or augmented virtuality environments is a cost intense task due to inherent complexity [Westera et al., 08]. Environments that combine flexibility and simplicity are rare. Especially, when it comes to the authoring of learning games, development costs are a serious issue. Also, authors of learning games often have a more educational background than a technical one.

*Research question (rq2):* Can we realize the above-mentioned architecture using freely available tools with open interfaces, while still reaching a quality level that allows creating attractive and useful educational games?

As stated in (rq1), we want to meet simplicity and flexibility requirements. *Simplicity* here refers to aspects like ease of use for players, understandability of the underlying concept and offered features, and practicability of the tools author game scenarios. Flexibility in our context refers to the ability to cover a broad range of possible usage scenarios with the same underlying concepts, architecture, and technical infrastructure.

To verify if we meet the simplicity requirement, we evaluated in a previous study a first mockup version of StreetLearn by presenting it to teachers with a cultural science background and to researchers from the field of technology enhanced learning. We asked them about "*their general interest in using such a tool for teaching, their foreseen ability to use the teacher tool to create a game scenario on their own, and their expectation towards student's acceptance and benefit of using StreetLearn-based games*" [Van Rosmalen, Klemke & Westera, 11]. Based on the positive but critical feedback we received, we extended the ARLearn architecture.

To verify, if our architecture meets the flexibility requirement, we identified a set of educational scenarios, which should be supported by ARLearn: Preparation/simulation/support or evaluation of physical trips, support for student exchange programs, simulation of specific situations and processes. We then applied ARLearn in three case studies to different educational scenarios using different client technologies. The outcome of these case studies is reported in this paper, after the description of the architecture.

## 4 ARLearn linking AR and AV

At the core of ARLearn lies the capability to extend an environment with virtual media. Media such as video, audio, open questions and multiple choice question can be bound to a location, time or game action. While some of these media serve the user with information, other media are intended to trigger reflection.

The ARLearn architecture has been complemented with two client applications. A Google Android application permits playing games in a real environment. A Google StreetView mashup lets users play these games in a virtual environment. Section 6.1 and 6.3 illustrates case studies that were conducted with Android clients, while section 6.2 details a StreetView case study. Furthermore, the ARLearn architecture supports mixed games. That is, games where both virtual players in StreetLearn interact with real players using the ARLearn Android client. However, although the toolset supports these mixed reality scripts, no case study has yet been conducted that illustrates these capabilities.

ARLearn features an open architecture as it can easily be extended with new features and enables third party applications to integrate with ARLearn by implementing XMPP based listeners. Extensible Messaging and Presence Protocol (XMPP) is a protocol for real-time messaging. ARLearn is an open source project [ARLearn, 11] and thus permits others to contribute to and reuse the code. The ARLearn architecture features a REST (Representational state transfer) API facilitating the creation of mash-ups enabling developers to interact with the ARLearn services. Having the ARLearn API running in the Google cloud does not require a local administrator maintaining the infrastructure. Moreover, teachers and students can use StreetLearn out-of-the-box and play StreetLearn games.

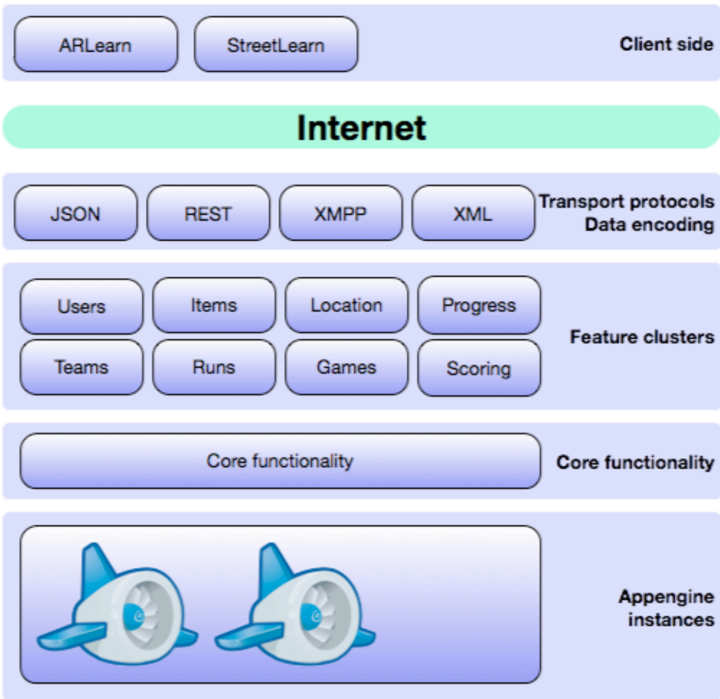


Figure 2: ARLearn software architecture

An ARLearn game is a blueprint for a serious game. Within a game, an author defines items, dependencies between items, game score rules and progress rules. Once a game has been created, an arbitrary amount of runs can be created and played. A run defines users grouped in teams. While users play a run, they generate actions (e.g. “reading a message”, “answering a question”) and responses. This output is also managed within the realm of a run.

The App Engine infrastructure is found at the lowest level of the ARLearn architecture. ARLearn builds on the JAVA version of Google App Engine using servlet, JDO (Java Data Objects), JCache and other JAVA technology. As a consequence, ARLearn will also run on standard JAVA servlet containers like Tomcat an Jetty. App Engine enables scaling up as the load on the infrastructure increases. It does so by dynamically creating new instances that take over part of the server load. The core functionality layer employs both a JDO interface to the App Engine Datastore, a schema-less object store and makes extensive use of JCache interface to the Google memcache. Google promotes extensive use of this high performant cache by providing an unlimited quota. Datastore operations are expensive and should be avoided through the use of the cache.

Two communication protocols for clients permit retrieving information from ARLearn. The REST based API features basic CRUD (Create, Read, Update and Delete) operations. For instance, a game can be created by posting an XML or a JSON representation to the service endpoint. However, for some operations it is important to notify the user instantly. For instance, if a game defines that all team members should get a message when a user picks up an item, simple REST communication is not a good candidate as it puts to much polling overhead on (mobile) applications. Practice has show that this has a negative influence on the battery lifetime and that this puts a load on the infrastructure that can be avoided. Therefore, an XMPP based notification system was implemented that only sends messages to the client when an event occurred.

In this notification system, each client maintains an XMPP connection and listens for messages coming from the App Engine server. As update messages should not conflict with a user’s chats, messages are not sent in plain text over this connection but are encoded as well-formed XML stanzas. The server broadcasts these XMPP messages for instance when a user’s score was updated because of an action of a team member, when a new item is visible or to update the location of team members.

## 5 ARLearn data model

At the core of ARLearn are media items that hold information or add a function to the game. In general, media items can be positioned on a map by providing latitude and longitude attributes. Not providing location attributes turns the item into message that users can receive at some point in the game. Through defining a multiple-choice item, users can answer questions with predefined answers. Audio feeds are integrated via audio objects with some explanatory text. Apart from information items, dynamic items such as a transport task let users perform actions. A transport task defines a pickup item and a dropzone. A pickup item can be taken by users and can be brought to a location where it is dropped. Dropping it at the correct dropzone can lead (through dependencies) to new available items, increased scores or increased game progress.

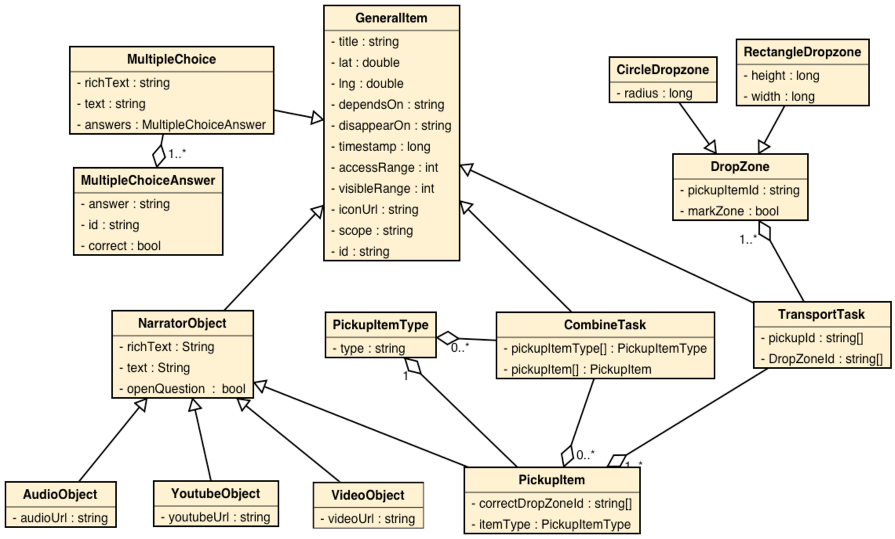

Figure 3: ARLearn object model

Generic items have a simple life-cycle with three states: At the launch of a run, an item can be either visible or invisible. When an item is invisible, it can become visible. Later, when the item is no longer needed, it can become invisible again.
Items can define a dependsOn and a disappearOn attribute to define the condition for the item to become respectively visible and invisible. A simple dependency mechanism is put in place to support these conditions.

- An *action-based* dependency becomes true once a certain game action has been triggered. For instance, the action “startRun” is triggered when the user starts a run for the first time. Alternatively, one can make an item depend on opening another item (“read” action). Scope can take three values: “user”, “team” and “all”. Making an item depending on a “read” action with scope all, will render the item visible as soon as one player in the run performs the “read” operation. Scope “team” will render the item visible if and only if a player in the same team opens the item and “user” requires the user to open the item first.
- A *time-based* dependency binds a time offset to another dependency. The dependency is fulfilled when two conditions are completed: the referenced dependency must be completed and the time must have elapsed with the amount of milliseconds specified by the offset attribute. For instance, if a message must be show to the user 10 minutes after the run has been started, a time-based dependency is created that refers to an action (startRun) based dependency with an offset of 600.000 milliseconds.
- Boolean dependencies provide a means to create “AND” and “OR” statements with other dependencies.

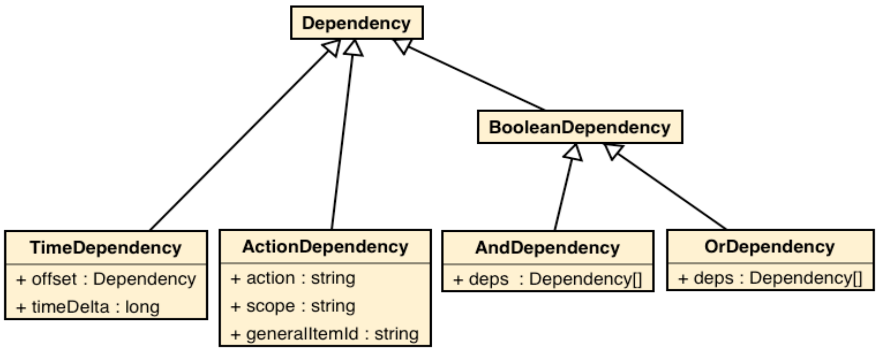


Figure 4: ARLearn dependency model

# 6 Case studies

In the following part we will present three case studies focusing on different challenges for linking AR and AV. We incrementally build the architecture to support different educational settings. Furthermore we designed the ARLearn architecture with two potentially conflicting goals in mind: simplicity and flexibility.

The following case studies illustrate the flexibility of the modelling approach and show how ARLearn can be applied to very different scenarios using a range of game patterns. For each of these cases, we briefly describe the scenario and its application as well as some lessons learned from each evaluation performed.

The three scenarios comprise:

1. In the Florence field trip case, an excursion script was created to support a city tour. It is implemented with a monitoring tool for the educator and using mobile devices with the ARLearn client installed in the physical environment with the learners.
2. In the Amsterdam Grachtengordel case, a police story in the drug scene is used to motivate cultural heritage learning content about Amsterdam. This game script follows an expository approach where learners remotely access the real world context via the desktop StreetLearn client.
3. The UNHCR hostage-taking intervention game models a hostage taking situation, that forces participants to act. The game process depends on players to take the right decision by using mobile devices with the ARLearn client the game is embedded in a real world situation and therefore quite authentic as a simulation.

These cases where selected with the goal to show the flexibility of ARLearn in mind. They differ with respect to their game design, their delivery channel, the immersive experience and the pedagogic approach (see table 1). The game scripts reflect a range of non-linear, location-based games (scavenger hunt), process-oriented location-based games (adventure game), and process-oriented event-based game (decision game).

These scripts cover the range of mobile games as introduced by [Nicklas, Pfisterer & Mitschang,, 2001]. The range of delivery channels chosen covers the mixed reality spectrum of Milgram's continuum [Milgram, 94]. With respect to immersion experience, we cover two of three immersion types as defined by [Dede, 09]: actional and symbolic immersion, leaving out the sensoric immersion, which would require a very different interaction and design approach and is thus left out in order to not conflict with the simplicity requirement. As [Taylor, 04] points out, developments in pedagogy move from transmissive (or expository) modes of teaching towards constructivist or socio-cognitive models (here referred to as situated learning). Within ARLearn, we are able to address both teaching models and it is in the responsibility of the game author to chose the appropriate model for the educational setting at hand.

Table 1: Comparison of game design, delivery channel, and pedagogic approach

| | *Florence case* | *Amsterdam case* | *Hostage case* |
|---|---|---|---|
| *Game design* | Scavenger game | Adventure game | Decision game |
| *Delivery channel* | augmented reality | augmented virtuality | augmented reality |
| *Immersion experience* | symbolic | actional | actional |
| *Pedagogic approach* | situated learning | expository learning | learning through decision taking |

As described above, in all three cases we performed small scale evaluations. We received encouraging comments, critical remarks as well as additional ideas as feedback from participants.

### 6.1 Florence field trip

**Scenario Background**

Every year, students of the School of Cultural Sciences take part in a field trip to Florence, where they study the available visual arts in the original context. During this trip, students train skills such as conducting a literature study, developing their own research questions and oral presentation skills. In the traditional setup of this excursion, the group is guided by a teacher through the city of Florence. There, at various locations, students have to instruct the group on a topic they prepared. The teacher intervenes in this process with questions.

Part of the group that visited Florence in the autumn of 2010 was equipped with the Android client of a first version of this framework. Via the smartphone, students received audio recordings containing either information or assignments relative to their location. The goal of this pilot was threefold. Firstly, we were interested whether a personalized learning experience was appreciated by the students. Secondly, with this toolset a paperless mobile field trip should be supported. Finally, an online portfolio should be transparently made available, enabling the learners to revisit their trip, but also to extend and further process the notes they made. The main advantage of this approach is that, unlike in the traditional setting, many students can work at once, while they are monitored by the instructor.


Figure 5: Map View (a), ListView (b), Item View (c) and providing an answer (d)

**Implementation**

In October 2011, this pilot was repeated with the more generic and enhanced ARLearn architecture. In this pilot, a sequential excursion was implemented through the dependsOn attribute. Eight participating students were each given a personal set of questions, recorded by the instructor. Four students were told that they would be actively monitored by the instructor, while the other four students would only get feedback after the field trip.

All questions appeared in a predefined sequence. Only after opening a question, the next question became available. The questions were presented in two views to the users. The gameplay was thus realized with the following object types:

- *AudioObjects* are used for all questions. Setting the "openQuestion" attribute to true enabled the users to respond to the questions.
- *Dependency* objects were only used to model the sequential order in which the questions must appear.

The map view shows the position of the questions, relevant to the user's location. The ListView (Fig 5(b)) lists all question ordered by the time they become visible. After opening an item, users were able to read the assignment, play the corresponding audio sample and answer the question. Students were able to use both spoken words ("Start recording", Fig. 5(d)) and pictures ("Add Picture", Fig 5(d)) to document their answer. Game-based elements, such as scoring and game progress tracking were not relevant here and were hence not added to the game blueprint.

**Evaluation**

After the pilot, the students and instructor were informally interviewed. During this interview, the following conclusions were drawn.

- The instructor liked this approach from a multitasking point of view. However, he had to invest a couple of hours in preparing the field trip. He was now able to remotely track the progress of several students at once, where in a traditional setting only one topic was covered at once. The instructor did express that real-time monitoring the progress (reading and interpreting the answers) of more than four students would not be feasible.
- During the 2011 pilot student interview all 8 students indicated they have no problem with the instructor tracking their progress. One of the 4 students that was monitored in real time, did call the instructor during the pilot to verify whether his answer was sufficient. This illustrates that students approached the tasks slightly different when they are conscious of being actively monitored. The group that participated in the 2010 pilot, had some students that voiced that they did not like to be monitored and suggested extending annotation with a feature to make them private.
- Students appreciated the ability to work on their own and synchronize their notes with the cloud where they could later revisit them.
- Students had mixed feelings with the map view (Fig 5(a)). Some students appreciated this feature while other students preferred the list view (Fig 5(b)) to lead them to their tasks. The nature of Florence (high buildings and narrow streets) often leads to poor GPS coverage causing problems with the location-based services.
- All students appreciated the sequential order of the tasks. Although, they could click through all assignments in advance, none of the students looked ahead.

### 6.2 Amsterdam Grachtengordel

**Scenario Background**

In September 2011 the Faculty of Cultural Sciences of the Dutch Open University initiated the development of a new cluster of courses in the field of cultural heritage. Google StreetView seemed to be an interesting medium for teaching in the field of public cultural heritage, such as (groups of) buildings or urban areas. For this case we chose a site from the UNESCO list of Dutch World Heritage (http://whc.unesco.org/en/list). The most suitable heritage site from this list was the Amsterdam canal ring area, because of its 100% coverage by Google StreetView. Within this area we made a diverse selection of buildings and subjects that would play a role in the *StreetLearn* game. The selection of buildings was done with the thought in mind that the diversity of the selection would be attractive for students and would illustrate the rich history of the area.

The storyboard for the game was based on real historic events: the smuggling of marihuana in fishing boats from Libanon to the Netherlands in the early 1970s. The game starts by introducing game scenario, learning objectives and the content scenario. The story involves fictive police officer Ada Jobse investigating in Amsterdam's drug scene in the late sixties. Her investigations take her around important cultural sites in the Grachtengordel, while tracking down drug smugglers and dealers. The player takes the role of Ada and has to move her around Amsterdam to interact with colleagues, drug users, dealers and smugglers to solve the case. On her way through Amsterdam, Ada gets to know many famous places and their historic meaning.

There is no direct relation between the content and the story of the game, but the tension between Amsterdam as a public museum on the one hand and a lively city on the other hand, is part of the content. The reputation of Amsterdam as a lively, tolerant and progressive city goes back to the 17th century, but during the roaring 1960s and 1970s this reputation got stronger than ever before. This is why it seemed to be interesting to give the gamer some information about this period. The drugs-story seemed adequate, firstly because the legislation of soft drugs stems from this period, and secondly via this story some couleur locale of the Amsterdam 1960s and 1970s could be presented.


Figure 6: Screenshot of the StreetLearn client in the Grachtengordel: Google StreetView with added interaction elements

**Implementation**

In the Amsterdam case study we used the StreetLearn client, which is a Google StreetView based user interface, that offers intuitive navigation and visualisation facilities and an open API for technical enhancements. The StreetView user interface is extended in StreetLearn with additional control & status elements (game controls) and visualisations of interactive game elements embedded in the 3D environment. On the backend side, the same technology is used as for the other two scripts: Google's App Engine provides the application server environment for the above mentioned architecture and object model.

The gameplay is realized in the object model with the following object types:

- *NarratorObjects* are used for all static information to be displayed. This comprises texts, videos, images, and links to external websites. NarratorObjects are used for two purposes: giving gameplay oriented information and delivering learning content.
- Objects of type *PickupItem* and *DropZone* are used to model search and delivery tasks (such as find a movie ticket and hand it in at the movie entrance to be admitted to watch a movie). This game pattern serves two main purposes in the game: on the one hand finding objects forces players to look around in their environment, while on the other hand, solving search and delivery tasks is an important gaming element to create tension.
- Using objects of type *MultipleChoiceTest* we can model interactive situations. These simulate dialogue situations, decisions, and they test the player's learning progress.
- *Dependency* objects help to model the story line. That way, some objects only become available after finding objects, delivering them, answering questions or even just after accessing specific information.
- *ScoringDefinition* objects and *ProgressDefinition* objects are used to model the game state: while progress objects model a linear game progress (indicating the percentage of milestones reached), the score objects can model the individual game performance.

**Evaluation**

We performed a half-day evaluation with 6 students and one tutor of the faculty of cultural science. Three of the players used the game in our lab environment, while the three other players where remotely accessing the game. The remote players where connected to us using a permanent Skype connection. We gave a short introductory presentation to the participants and then asked them to play the game. During the game we gave some support to participants and answered questions.

After the game session, we asked participants to fill out a questionnaire and we additionally collected some feedback in a wrap-up discussion. Four of the students were female, two were male. The participants where in age groups 36-50 (2 participants), 52-65 (2 participants) and above 65 (2 participants). Their previous experience with technology enhanced learning, computer games and serious games was rather low (avg. 2.5 on a scale from 1 to 6, where 6 is “I fully agree”).

**Evaluation results**

**Concept.** The participants rated the use of serious games for education as helpful (avg. 5/6) and the idea behind StreetLearn as interesting (avg. 5/6). However, some of them claimed, that learners should concentrate on learning rather than gaming (avg. 3.3/6)

**User Interface.** Participants were asked to rate the user interface of the StreetLearn prototype (on scales of 1 to 5, where 5 is "I fully agree"). They rated it to be appealing (avg. 3.5/5), partly easy to use (2.5/5), and useful for the game (3.5/5), only partly they think, the UI is obstructing (2.5/5), too complex (2/5), or confusing (2/5).

**Learning content.** The learning content about the Amsterdam Grachtengordel was rated as interesting (4.2/5), relevant (4/5), understandable (4.2/5), nicely presented (3.8/5), comprehensive (3.5/5). Only few participants rated the content as too complex (2.2/5).

**Game story.** The game story in the Amsterdam drug scene was perceived to be well connected to the learning content (4/5). However, opinions on the usefulness of the game story where not clear: the story was rated only by half of the participants as interesting (2.8/5), motivating (2.7/5), helpful (2.7/5). It was also by half of the participants rated as confusing (2.8/5), distracting (2.8/5), or too simple (2.7/5).

**Evaluation summary**

Overall we are satisfied with the outcome of the evaluation. Participants were highly interested and motivated. We received very positive feedback for the concept and idea behind StreetLearn and some critical remarks about its current prototypical status. The feedback given is very helpful for further improvements to the system.

Confronted with the game, it became clear that the content was too complicated. The starting point of the game is the canal ring area, which is a 17th century construction. But then the selected buildings, except for the Westerkerk, date from other centuries. Also the relation between the drugs-story and the World Heritage-status of Amsterdam was not completely clear. The attempt to make the content diverse and refreshing, turned out into an unforeseen complexity.

The six students who tested the game, came to the same conclusion. In general, they were positive about the game, especially about: 1) the content (they even wanted more information and more multiple choice questions); 2) the diversity of the media, such as photos, texts, audiovisual material; 3) the use of StreetView; 4) the fun of playing the game; 5) the connection of the content with the present time. Their major critical remarks about the content were the following: 1) the connection between the canal ring area and the drugs-story is not clear; 2) the drugs-story is distracting, childish and gives a too stereotypical image of Amsterdam; 3) the clues how to go on were too difficult.

For this specific educational purpose, teaching about buildings, their architecture and history, there is another aspect of StreetView that must be criticised here: the fact that in StreetView it is hard to focus on one specific building and see it completely, without distortions.

**Conclusion**
Two important conclusions can be drawn from this case. First: when using Google StreetView and a game for teaching, the content and the story should not be too complex. There is already enough distraction in StreetView as such and presenting the content in the form of a game splits the information in fragments. This way of presenting content is refreshing, but if the content in itself is too diverse, the gamer gets confused. Second: the selection of buildings should not only be based on the (art)historical importance, but also on the quality of the photographic image of the site in StreetView. Because of the difficulty to focus on one building, maybe one should take groups of buildings or, in this case, a whole canal.

### 6.3 UNHCR hostage taking game

**Scenario Background**
The Office of the United Nations High Commissioner for Refugees (UNHCR) leads and co-ordinates international action to protect refugees and resolve refugee problems worldwide. As this organisation is sometimes confronted with kidnappings of their co-workers, employees are trained on how to deal with such situations. An ARLearn "decision making" game was designed that presents the participants a real-time simulation of a hostage taking situation. The game script was created taking into account several roles (Head of Office, Security Official and Staff Welfare). Depending on the role, participants receive different tasks and information. For instance, the head of office receives calls from journalists, while staff welfare receives a call from a distressed hostage's family member. Therefore different educational scenarios and collaborative scripts have been implemented in ARLearn to simulate complex hostage taking scenarios and their management with different roles.

The objective of these scripts was threefold.

1. UNHCR aimed at giving their trainees an authentic learning experience. An incoming video message with a plea for help from the hostage created a sense of immersion.
2. Personalising the game script to the role of the trainee enabled multitasking. The hope was to have one operator who could easily manage several games. This goal was only partly met as it was observed that manually controlling and intervening with 9 devices was a limit.
3. Finally, through overloading the participants with many messages and tasks, the game designers wanted to create a level of stress. For this purpose the possibility to trigger notifications automatically, was extended with the possibility for a game operator to trigger them manually. This way, the operator can better estimate when a message (with additional work) should be dispatched.

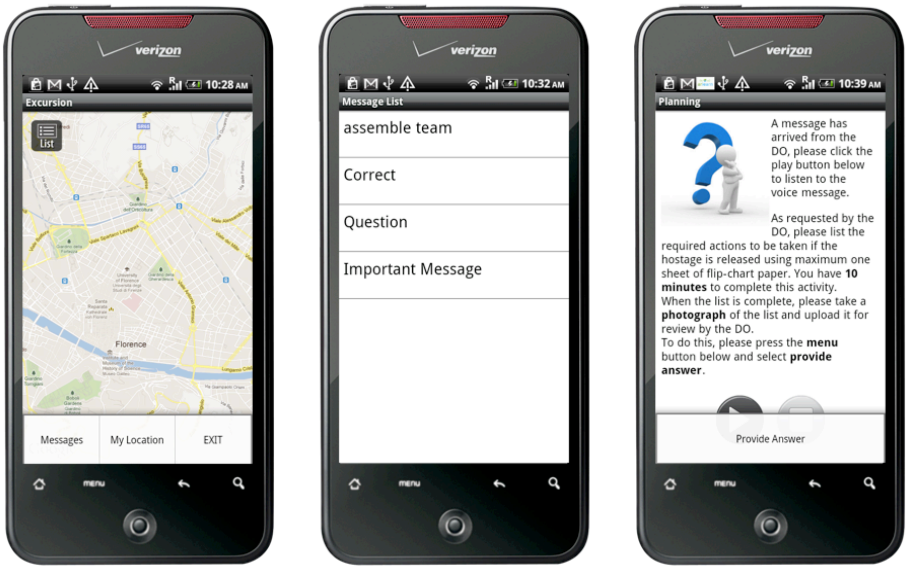


Fig 7: UNHCR screenshots: MapView (a), Message View (b), example assignment

**Implementation**

This game script was implemented in two phases. In November 2011, a dry-run was organised in Budapest with staff members of the organisations. During this dry-run it became clear that the Internet connection is a bottleneck. Having 6 devices connected to a hotel's Internet hotspot produced network interference. In December 2011, the actual pilot was organised in Entebbe, Uganda. Here, 3 game runs were ran at the same time featuring 3 roles per run. So 9 devices were used in parallel, connected via a cellular network. Rather than relying on a single local wifi, the 9 devices were connected to the Internet through the cellular network, which resulted in a better Internet connection.

The gameplay is realized in the object model with the following object types:

- A *VideoObject* with a plea for help was played at the start of game.
- *NarratorObjects* are used to broadcast information or to simulate an incoming call by a journalist (*AudioObject*). In some cases for instance participants had to prepare and send in a document. This was modelled with the open question attribute that allowed the participants to answer with a picture of the document.
- At many stages in the game, objects of type *MultipleChoiceTest* allow players to make decisions. E.g., giving three options on how to react on a journalist's call.
- *Dependency* objects help to model the decision making structure. When giving an erroneous answer, players receive contextualized feedback. Dependencies were also used to synchronize all players. E.g., all players receive after approximatly 15 minutes a message from the DO to assemble in

the control room. Here an action is manually fired by the operator, to trigger this message on all devices.The operator was given a simple dashboard to trigger these manual actions.

**Evaluation**

Although no summative and quantitative evaluation was organised for this pilot, the game organiser provided the following formative feedback.

- Being able to run the game on Android devices has many advantages. Although this was not the case for the UNHCR pilot, ARLearn games can be ran on personal devices, implementing a “bring your own device” (BYOD) strategy. Leverage a widespread operating system, makes the solution much cheaper compared to AR solutions that need special hardware. In this particular case however, learners had to become familiar with the device (touchscreen, Android OS, etc.). Therefore it was suggested to create a demo run to enable participants to get accustomed to the device.
- The interface of the Android application was centred around a map. In the UNHCR case no content was associated with coordinates. Game logic and notification was only relying on the message timing and tasks assigned. For these kind of scripts the map was sometimes confusing participants.
- The script was implemented with both manual and automatic triggers for items to appear. Although manual triggers offer some degree of flexibility, future pilots should have more automatic triggers and fewer dependencies on a network connection. The unreliable network results sometimes in manual triggers not arriving at a device. Automatic triggers have the advantage that they are cached on the mobile device and they lower the work load for the operator.
- Lowering the dependency on Internet connection will make the game easier to port to other countries, not having to acquire many sim cards or deal with wireless settings (e.g. some essential network ports being blocked by the firewall of a hotel).

### 6.4 Results

With the application of our ARLearn architecture to three different cases, we were able to demonstrate its flexibility with respect to game design, delivery channel, and pedagogic approach.

**Game design.** The game design underlying the three different cases differs largely. While the Florence field trip represents an explorative location-based game in the style of a scavenger hunt, the Amsterdam case combines a rather linear game design based on a police story with location-based learning content. The hostage case represents a decision game, where players have to take decisions and learn their consequences during the game play. All three cases could be modelled with the object model presented above.

**Delivery channel.** While the Amsterdam case uses Google's StreetView technology on stationary devices to deliver game interaction and learning content as an augmented virtuality game, the other two cases where delivered via mobile devices and the use of augmented reality enhanced user interfaces. In all three cases, the same backend architecture has been used. This demonstrates the flexibility of the architecture to deliver the game experience to both, physical and virtual environments.

**Pedagogic approach.** The pedagogic approach, which motivated the three cases, also differs significantly. In the Florence case, the learning content is distributed to allow an exploratory learning approach. Students should be enabled and motivated to explore an unknown city. The game environment offers information at specific locations, which participants can discover and explore. In the Amsterdam case, the game story offers a rather linear frame which guides the player around different locations in Amsterdam. At the different locations players have to solve tasks, find objects, and retrieve information, which comprises game-related information as well as learning content. The game story thus represents a motivator for the player to access learning materials, which are offered in a rather expository way. In the hostage case, learning and game-design are closely connected: a player is confronted with a series of decision situations. Decisions taken lead to consequences and feedback presented to the player results in subsequent decision situations. Learning takes place as part of the decision/feedback/consequence process. Also with respect to the pedagogic approach we could thus show the flexibility of the chosen approach. ARLearn offers the author of an educational game the freedom of choice to create game scenarios, which match the pedagogic requirements. Flexibility here refers to the possibility to define the level of freedom the learner will have: from strictly linear, expository processes to exploratory, situated learning settings, including mechanisms to define decision points.

## 7 Future work and conclusions

This manuscript presents an open source, extensible architecture and mobile tools that are realized with freely available tools and offers an open REST API. From the end-user point of view, playing games is easy for users and requires no special knowledge. Creating scripts requires no programming skills but does impose still technical background as scripts are to be edited either in JSON or XML.

The promising results of the first three application cases of the ARLearn architecture motivate us to perform further research and development activities, that lead to extended applicability of the architecture. We believe that the ARLearn approach offers a promising framework to setup immersive learning environments with relatively low effort. Therefore it will become easier to analyse their effects on inert knowledge and far transfer as discussed in the introduction of the article. For further developments we are planning to explore the following directions:

**Mixed reality games.** While we have shown applicability in augmented reality and augmented virtuality scenarios, we did not yet explore a scenario where both approaches are mixed. We envision game scenarios where teams can be composed of mobile players and players that use stationary devices.

**Indoor scenarios.** GPS navigation and StreetView coverage have limitations as soon as scenarios involve indoor navigation and visualisation. However, as localisation technologies for indoor navigation become available for mobile devices and as Google starts to include interiors of buildings into StreetView, indoor scenarios may be better supported by future versions of ARLearn.

**Authoring environment.** Within the current architecture we defined a service-based interface for the creation of all objects required to define games, runs, and contents. The technically most simple way to utilise this interface is the creation of initialisation scripts. However, this way of authoring games is not suitable for persons without technical backgrounds. Therefore, we aim for the creation of authoring tools, which use the same range of user interfaces as the game itself to offer authoring functionality: augmented reality and augmented virtuality tools.

**Connectivity.** When organising field trips on different locations, connectivity is often an unreliable component. Other parties with an interest in the ARLearn mobile client have already listed the requirement for a network independent version that would be prepared and loaded to the device prior to the field trip.

**Maps**. The current version of the Android app relies on the Internet to update its maps and prohibits the software to cache the Google Maps in advance. Therefore, future work will focus on building on OpenStreetMap, an initiative that supports caching maps in advance empowering offline usage of the toolset.

## References


[Appelman, 05] Appelman, R.L. (2005). Experiential modes: a common ground for serious game designers. International Journal of Continuing Engineering Education and Life Long Learning. 15(3) pp 240-251. Inderscience, 2005

[ARLearn, 11] ARLearn project page. (2011) http://code.google.com/p/arlearn/

[Bransford et al., 90] Bransford, J. D.; Sherwood, R. D.; Hasselbring, T. S.; Kinzer, C. K.; Williams, S. M. (1990). Anchored Instructions: Why we need it and how technology can help. In Nix, D.; Spiro, R. (eds.). Cognition, Education and Multimedia: Exploring ideas in high technology (pp. 163 - 205), Hillsdale, NJ: Erlbaum.

[Brown & Cairns, 04] Brown, E. & Cairns, P. (2004). A grounded investigation of game immersion. CHI ’04 extended abstracts on human factors in computing systems. (pp. 1297-1300). New York: ACM.

[Dede, 09] Dede, C. (2009). Immersive interfaces for engagement and learning. Science, 323(5910), 66-9. AAAS. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/19119219

[de Freitas, 06] de Freitas (2006). Learning in immersive worlds, A review of game-based learning.

[Doswell and Harmeyer, 07] Doswell, J., Harmeyer, K. (2007). Extending the 'Serious Game' Boundary: Virtual Instructors in Mobile Mixed Reality Learning Games. Digital Games Research Association International Conference (DiGRA 2007), Tokio, Japan (2007)

[Herrington, Reeves & Oliver, 07] Herrington, J., Reeves, T.C., and Oliver, R. (2007). Immersive learning technologies: Realism and online authentic learning. Journal of Computing in Higher Education. 19 (1), 65-84.

[Klopfer, Osterweil & Salen, 09] Klopfer, E., Osterweil, S. and Salen, K. (2009) Moving Learning Games Forward, Obstacles Opportunities & Openness, Cambridge MA: MIT/The Education Arcade, [Online], Available: http://education.mit.edu/papers/MovingLearningGamesForward_EdArcade.pdf [2 April 2011]

[Kolb, 84] Kolb, D. A. (1984). Experiential Learning. Englewood Cliffs.

[Lave & Wenger, 90] Lave, J., & Wenger, E. (1990). Situated Learning: Legitimate Periperal Participation. Cambridge , UK : Cambridge University Press.

[Liestøl, 11] Liestøl, G. (2011). Situated Simulations between Virtual Reality and Mobile Augmented Reality: Designing a Narrative Space. In Furht, B. (Ed.) Handbook of Augmented Reality, pp. 309-319. Springer.

[Milgram and Kishino, 94] Milgram, P. and Kishino, A. F. (1994) Taxonomy of Mixed Reality Visual Displays IEICE Transactions on Information and Systems, E77-D(12), pp. 1321-1329.

[Nicklas, Pfisterer & Mitschang, 2001] Nicklas, D., Pfisterer, Ch., Mitschang, B. (2001), Towards Location-based Games. In: Proceedings of the International Conference on Applications and Development of Computer Games in the 21st Century: ADCOG 21. Hongkong Special Administrative Region, China, pp. 61-67.

[PWC, 10] PWC (2010) Global Entertainment and Media Outlook: 2010-2014. [Online], http://www.pwc.com/

[Santamarina et al., 10] Santamarina, R.T, Torrente, J., Moreno-Ger, P., Fernández-Manjón, B.: e-Training DS: An Authoring Tool for Integrating Portable Computer Science Games in e-Learning. 9th International Conference on Web-Based Learning (ICWL 2010), Shanghai, China, 8-10 December 2010, pp. 259-268. 2010

[Specht, Ternier & Greller, 11] Specht, M., Ternier, S., & Greller, W. (2011). Dimensions of Mobile Augmented Reality for Learning: A First Inventory. Journal Of The Research Center For Educational Technology, 7(1). Retrieved January 18, 2012, from http://www.rcetj.org/index.php/rcetj/article/view/151

[Taylor, 04] Taylor, J. (2004). A task-centred approach to evaluating a mobile learning environment for pedagogical soundness. In: Attewell, Jim and Savill-Smith, Carol eds. Learning with mobile devices: research and development. London, UK: Learning and Skills Development Agency, pp. 167–171.

[Van Rosmalen et al., 08] Van Rosmalen, P., Sloep, P., Kester, L., Brouns, F., de Croock, M., and Pannekeet, K. (2008) ‘A learner support model based on peer tutor selection’, Journal of Computer Assisted Learning, vol. 24, pp. 74-86.

[Van Rosmalen, Klemke & Westera, 11] Van Rosmalen, P., Klemke, R., & Westera, W. (2011) Alleviating the Entrance to Serious Games by Exploring the Use of Commonly Available Tools. In Proceedings of the 5th European Conference on Games Based Learning, 20-21 October, Athens, pp. 613-619.

[Westera et al., 08] Westera, W., Nadolski, R., Hummel, H. and Wopereis, I. (2008) ‘Serious Games for Higher Education: a Framework for Reducing Design Complexity’, Journal of Computer-Assisted Learning, vol. 24, no. 5, pp. 420-432.