Supporting Crisis Simulations with the ARLearn Toolkit for Mobile Serious Games

Stefaan Ternier, Atish Gonsalves, Bernardo Tabuenca, Fred de Vries, and Marcus Specht [*]

Introduction

Although the transfer of factual knowledge plays an important role when training personnel who operate in potentially dangerous conditions, mobile learning techniques to support security training are often underused. There is a great value in simulating dangerous situations in the training context. In contrast to learning from a textbook, a simulation leaves room for customization and surprise, as players do not know what will happen in advance. By experiencing a series of events that occur in a simulation, participants learn in an authentic context and are trained to respond to events as they occur. This is different from studying factual knowledge. The theory of situated learning supports this assumption, and states that learners do not learn via the plain acquisition of knowledge but rather through active participation in social contexts.[1] Organizing simulations in the real world can become very expensive and resource-intensive. For instance, a real-life military simulation can require the use of weapons and explosives to engage the participants in the scenario. Many dangerous situations require personnel with different roles (e.g., team leader, communication expert) to cooperate. Managing these different roles during a simulation requires extra work of the simulation facilitator.

One of the challenges of organizing a simulation virtually is to implement an immersive experience. Chris Dede defines "immersive learning" as learning that involves the "subjective impression that one is participating in a comprehensive, realistic experience."[2] Such experiences can be typically found in games and simulations. Dede differentiates between the following types of immersion:

- *Sensory immersion* refers to the possibility to get haptic feedback via sensors and to create a realistic environment.
- *Actional immersion* makes it possible for the user to take actions that would normally be impossible in the real world. Such actions can lead to new insights.
- *Symbolic immersion* focuses on making new semantic and psychological associations that are acquired via the content available in a virtual environment.

[*] Dr. Stefaan Ternier, Bernardo Tabuenca, Fred de Vries, and Prof. Dr. Marcus Specht are from the Open University in The Netherlands and Atish Gonsalves is from UNHCR.

[1] Jeanne Lave and Etienne Wenger, *Situated Learning: Legitimate Peripheral Participation* (Cambridge, U.K.: Cambridge University Press: 1990).

UNHCR seeks applications of new technologies to train its staff in a number of key subject areas, such as refugee law, international human rights law, operations planning, and security risk management in order to protect and meet the needs of humanitarian agencies and forcibly displaced persons. Some of these topics can be delivered through conventional e-learning delivery methods, such as self-study modules and webinar-based coaching. However, training humanitarian staff for complex scenarios, such as managing hostage situations, relies more on immersive and participatory simulation exercises and role-playing activities. For the application of mobile devices in a role-playing game, we are particularly interested in actional immersion. By enabling the participants to make decisions during a crisis simulation, the simulation will follow a different course, taking into account the decisions the participants make in particular situations.

Mobile devices play an important role in crisis simulations that involve role-playing. Although they do not match the immersive feeling a participant gets when entering a real-world simulation (e.g., flying a full-scale flight simulator), mobile devices are very well suited to orchestrate a flow of events. In a mobile serious game, the mobile device acts both as a remote control and a display. The device offers the participant a perspective to the game that is properly adapted to the user's role and context. At the same time, the device is used to provide the game with input. Users record audio and video with their smartphone, provide answers to open and multiple-choice questions, or make the game controller aware of their current context (such as their current location).

ARLearn is a toolkit for designing and playing mobile serious games, and is especially useful for mobile scenarios including field trips and role-playing games. The combination of a serious game engine with a mobile platform has produced a toolkit with significant potential for simulating crisis situations. The Open University of The Netherlands (OUNL) and the UNHCR have collaboratively developed a serious game to simulate hostage situations with the ARLearn application, and have trained UNHCR staff with it.[3] The next section of this essay covers the ARLearn features and software architecture, and how they support a mobile simulation. The following section includes a UNHCR case description that illustrates how the application has been used in different educational scenarios in which learners are to explore their environment and collect data, commonly addressed by the concept of fieldwork.[4]

ARLearn Toolkit

ARLearn, originally a tool for audio augmented reality, has grown from a standalone smartphone app into a fully-fledged mixed reality application platform including sup-

[2] Chris Dede, "Immersive Interfaces for Engagement and Learning," *Science* 323 (2009): 66–9; available at www.ncbi.nlm.nih.gov/pubmed/19119219.

[3] "ARLearn Project Page," 2011; available at http://code.google.com/p/arlearn.

[4] Stefaan Ternier, Roland Klemke, Marco Kalz, Patricia van Ulzen, and Marcus Specht, "ARLearn: Augmented Reality Meets Augmented Virtuality," *Journal of Universal Computer Science* 18 (2012): 2143–64.

port for field trips, serious gaming, augmented virtuality, and notification systems.[5] In order to support the creation of simulations, ARLearn builds on two important concepts:

- A *game* is a blueprint for a simulation, and defines the game artifacts, user roles, and the logic that combines these artifacts.
- A *run* refers to a concrete play of the game and defines users grouped in teams. Within a run, the actions of these users, their responses and progress are tracked. For each game, an arbitrary amount of runs can be created.

ARLearn implements a simple data model that enables the definition of several kinds of media artifacts, including multiple choice messages, video messages, and audio messages. Media artifacts are bound to a context that can be defined by a location and/or a timestamp. The context defines where or when in the game messages have to appear. Furthermore, a flexible dependency mechanism enables the author to define the game logic. Through dependencies, the author specifies conditions for making media artifacts appear or disappear. An author can, for instance, define that sixty seconds after all players have read the introduction message, a video message will become available. Media artifacts can be configured so that they will disappear about three minutes after the artifact has been read.

Dependencies

Dependencies enable a game author to define the game logic. An ARLearn media artifact defines two dependencies:

- When an *appearOn* dependency is specified, the media artifact is not visible when the run starts. Only after the dependency is satisfied is the media artifact rendered visible.
- The *disappearOn* dependency defines when an artifact will become invisible.

Dependencies do not evaluate to Boolean values ("true" or "false"), but rather evaluate to a timestamp when the condition is satisfied. If the system cannot determine when the dependency will satisfy, the dependency evaluates to an infinite number (i.e., the biggest number that can be represented by the system). Evaluating to a timestamp enables authors to specify time constraints and allows the mobile client to schedule a timer. ARLearn is currently equipped with three kinds of dependencies:

- An *action-based dependency* refers to a game action. Every message, for instance, specifies a "read action." This action is triggered a soon as a user reads the message. Open questions or multiple-choice questions trigger an action when a response has been given.
- A *time-based dependency* specifies a time relative to another dependency. With this dependency, one can stipulate the time that must pass after an action

[5] Ibid.

has occurred. This mechanism allows for encoding a natural sentence like "*thirty seconds after the artifact has been read.*"

- *Combined dependencies* (AND/OR) combine two or more dependencies. A game author can thus use expressions to specify multiple conditions that need to be fulfilled. An AND-based dependency will evaluate its operands and will return the maximum timestamp. An OR-based dependency does the same, but returns the lowest timestamp.

These dependencies support flexible game logic. Take the example of defining a countdown for answering a question. Consider the case where a player is given three minutes to provide an answer. If the player does not succeed to answer within that time, the question disappears. Such a countdown clock is implemented via a disappearOn dependency. This combines a time-based and an action-based dependency to express a predicate that specifies "180000 milliseconds after the action 'answer_given' has been issued for the question." As long as the action has not been triggered, the action-based dependency will evaluate to an infinite amount of time. The time-based dependency adds three minutes to this value.

Notifications

Notifications are a central element of ARLearn, both from a technical and conceptual point of view. In a typical multi-player game, players will receive messages as a result of actions taken by other players. For instance, as soon as the first responder arrives on a scene, all other players receive a message that informs them. The ARLearn game engine will send notifications to all players when a media item becomes visible.

Technically, ARLearn supports three mechanisms to deliver notifications to the player's device. Efficiently organizing notifications was an important achievement, since energy-efficient mobile devices cannot continuously poll for newly available media artifacts.

The Google Channel application programming interface (API) is a JavaScript solution for sending asynchronous notifications (so-called long polling). With the channel API, an application client subscribes to a common channel and listens for updates. This solution typically works well in a browser, but is less suited for mobile devices because the solution is not energy efficient, as it requires the app to run its own notification channel as a background service. Furthermore, long polling requires the client to be continuously online. Typically, mobile users can enter and leave networked places and therefore need an application that is optimized for offline usage.

Apple Push Notification (APN) and Google Cloud Message (GCM) deliver messages to iOS and Android devices respectively. Both technologies rely on one communication channel that the device maintains for all applications, and is thus much less resource-intensive. The ARLearn client for iPhone relies on APN, while the Android app is currently being migrated from the Google Channel API to GCM. The ARLearn authoring environment, a JavaScript browser application, features a module to track the user and to be notified immediately when new responses are produced. This module thus builds on the Google Channel API.

Roles

Media artifacts can be bound to roles, meaning that they will only be visible to players that have the same role assigned. For instance, a message "answer an incoming call from journalist" can be bound to the role "communication officer." Content that is not bound to a role is visible to all players. Dependencies can be used in combination with roles. This is useful when content needs to appear or disappear when a player in a given role performs an action. This enables expression actions like "make an information package available to all users, when the team leader enters the control room."

Tagging and Scanning

Often, mobile serious games require actions that are bound to a user's location. Although a GPS sensor comes in handy here, a very useful alternative is checking in to a location by scanning a tag. ARLearn supports scanning both QR codes and RFID tags. QR codes require the user to actively open the scanner to scan the tag by pointing the camera to it, while an RFID tag only requires touching it with the phone. The latter does not oblige the user to open the scanner for scanning the object, an action that relies on more user interaction cycles with the mobile app. The advantage to scanning over automatic recognition of a user's location via GPS is twofold. Firstly, GPS is often not reliable. In places with tall buildings, reflective windows, and narrow streets, GPS signals are often very weak. When users are indoors, the signal is too inaccurate, even if used in combination with network-based location tracking. Secondly, allowing the user to scan a tag to reveal his location puts the player in control. The player actively decides to inform the system that he entered a room, and gets direct feedback that this action was registered.

The UNHCR Hostage-Taking Pilot Project

UNHCR operates in over 120 countries, and its staff members often work in hazardous locations. To better equip its staff, UNHCR conducts a number of security management training programs worldwide. The UNHCR Global Learning Center (GLC) in Budapest organizes these trainings. The typical approach is a workshop organized over a three to five-day period in which different aspects of security risk management are addressed. These workshops cover policy-based information, such as standard operating procedures, delivered through self-study e-learning modules done in advance of the course, followed by instructor-led workshops. The workshop also includes immersive simulation exercises, for example hostage taking, bomb threats, and other security-related scenarios.

For many years a role-playing game has been part of these workshops. In this game learners are split into groups representing the different roles that are needed in a real security situation. As a team, each group implements and carries out the procedures that have been introduced during the e-learning modules and the workshop. In the case of the hostage-taking scenario, a role-playing game is a highly immersive experience for the learners, in which they have to manage stress, act quickly, collaborate and negotiate in order to "save the hostage." Running the game is an intensive exercise, not only for the participants, but also for the organizers and facilitators. It requires a lot of concentration and effort by a facilitator to encourage the whole team to engage in the chaotic development of a hostage situation. This limits the number of participants that can simultaneously join a game. Since the game is carried out at a rapid pace, it can be difficult to have an all-inclusive debriefing in which all roles on all teams receive appropriate feedback. The debriefing and reflection phase of the activity is a major learning point. Debriefing allows learners to reflect on what they learned, the challenges and risks associated with hostage situations, and their personal capabilities. Facilitators recognize that this is one of the most important learning moments, but it is frequently difficult to capture all the important points from the simulation due to the speed at which events occur. This aspect has led to the development of this project, in an effort to address this shortcoming.

As an alternative to the original game, a version was developed by applying concepts of mobile serious games and blended instructional design principles.[6] Using the ARLearn toolkit made it possible to address the following training issues:

- Enabling the creation of different reusable variations of a game design for emergency security response, covering initially the hostage situation and potentially other cases
- Enabling "on the fly" messaging to participants and real-time assessments of activities
- Semi-automatic management of the game, thus enabling more participants to experience the role-playing exercise.
- Creating a log of responses and interactions throughout the game, which can be used by the trainer to provide feedback during the debriefing session.

Game Design

The first ARLearn pilot in UNHCR was implemented at the Security Management Learning Program (SMLP) workshop in December 2011 in Entebbe, Uganda. The seventeen participants were senior staff members, including heads of offices, who are responsible for managing operations and ensuring the security of the staff members in their country offices.

A pre-game survey was conducted to examine how comfortable the participants were with smartphones. The majority used their mobile phone frequently, either constantly (nine of the seventeen) or daily (six). Eleven participants used their phone to access the Internet and email. Six of them constantly accessed the Internet with their smartphones. This survey showed that most of the participants were comfortable with a

[6] Marion Gruber, Christian Glahn, Marcus Specht, and Rob Koper, "Orchestrating Learning Using Adaptive Educational Designs in IMS Learning Design," in *Sustaining TEL: From Innovation to Learning and Practice*, ed. Martin Wolpers, Paul A. Krischner, Maren Scheffel, *et al.* (Berlin, Heidelberg: Springer Verlag, 2010).

smartphone, although most of them had not used a touch-screen based device prior to the simulation.

The workshop was organized in the conference room of a hotel in Entebbe. Internet connectivity was limited and sporadic, and electricity cuts were quite frequent. In Entebbe, the game was played using nine smartphones simultaneously. Three runs were created for the game, with three roles participating in each run. Each role was assigned to two or three players.

The game is designed to train the participants on the response procedures to be initiated immediately when a staff member is taken hostage. A Hostage Incident Management (HIM) team is deployed eventually in such situations, but it can take time for this team to arrive, and country offices need to know how to respond prior to their arrival.

The players participated in this game taking one of the following three roles: head of office, security officer, and staff welfare officer. The hostage-taking simulation was designed so that players in all roles play the same game but have to react differently based on their roles. The game is organized in five phases:

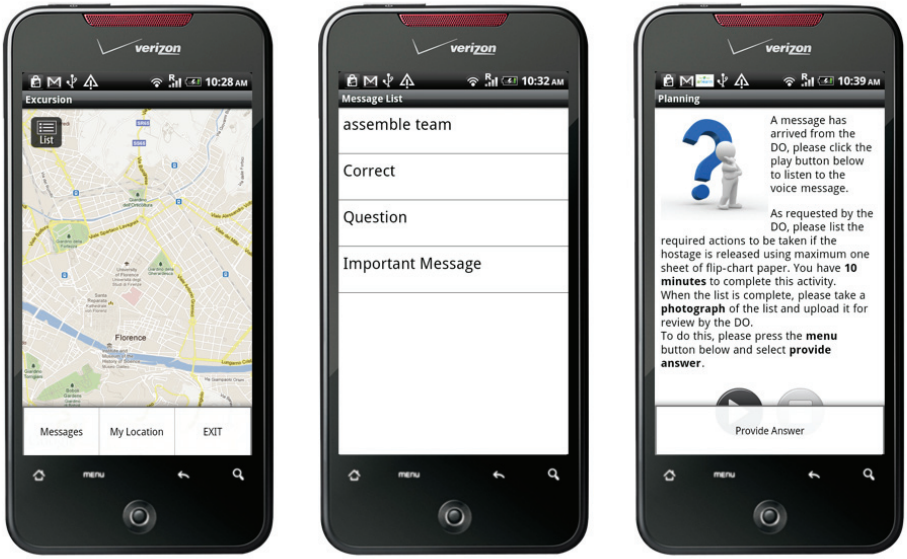
Phase 1: Notification of the Incident. The game starts with a plea for help by Jerry Khan, a fictitious UNHCR employee who has been taken hostage. This video message features a blindfolded actor and creates an authentic context. This message is broadcast to all the roles. Next, players make a decision on what to do next, depending on their specific role. The head of office (Role A), for instance, can decide to "notify the Designated Officer (DO)," while a staff welfare officer (Role C) should select the option to "contact senior management." Depending on the decisions they make, they receive feedback on whether these were good choices.

Phase 2: Assembling the Team. In the next phase, the head of office is informed by the DO that a hostage incident management team will be dispatched. In the meantime, they need to contact the security advisor (Role B) and staff welfare officer (Role C) and ask them to assemble in headquarters for a planning session.

Phase 3: Planning. When the facilitator observes that the team has assembled, an audio recording of the DO requesting the team to work out a reception plan is sent out. The team is next assigned to work out this plan on a flip chart and to capture a photo of the plan with their mobile device and submit this as soon as they are ready. Next, the participants are asked to split up and go to their individual rooms.

Phase 4: Responding. In this phase, Role A and Role C participants are to respond to calls from a journalist and a distressed family member, respectively. Meanwhile, the security officer (Role B) receives a message from the DO with the task to prepare a Proof of Life (POL) question.

Phase 5: Negotiating. In this last phase, all roles gather together again. This is triggered by a message from the hostage takers. In this phase, a negotiation with the hostage takers is simulated. The game ends with the message that the Hostage Incident Management (HIM) team has arrived and is ready to take over the negotiations.



assemble team

Correct

Question

Important Message

A message has arrived from the DO, please click the play button below to listen to the voice message.

As requested by the DO, please list the required actions to be taken if the hostage is released using maximum one sheet of flip-chart paper. You have **10 minutes** to complete this activity.
When the list is complete, please take a **photograph** of the list and upload it for review by the DO.
To do this, please press the **menu** button below and select **provide answer**.

Figure 1: UNHCR Screenshots, from Left to Right: Map View, Message View, Sample Assignment.

Results

Overall, the game was well received, and the participants were able to successfully complete the runs. During the run, there were three electricity cuts that caused the wireless router to reset. However, the participants were able to continue the exercise despite these challenges.

The participants found the ARLearn simulation very useful, and once they got over the initial technical obstacles were able to respond to the Notification, Assembling, Planning, Responding, and Negotiating exercises. Of the seventeen participants, fourteen answered positively on the question regarding the usefulness of mobile phones in a simulation exercise (one response was negative, and two were mixed). As both the exercise and the use of smartphones in training was new to the participants, the expressed learning outcomes varied, from fourteen persons learning from the game itself, to two people expressing that they mainly learned about using the smartphone, and one reporting no learning.

From a content perspective, the participants were able to understand the critical nature of the exercise and the tasks associated with the immediate response of a hostage-taking scenario. They learned the importance of coordination, which was enabled and assessed through the game. The game added more realism through the "Jerry Khan" hostage video, calls from distressed family members, and pressure that was applied by the demands of the hostage takers. In addition, the facilitators played a role in moderating and pacing the flow of the game through the use of the manual triggers. It created good inter-dependencies among the roles, and showed that the leadership of the head of office (Role A) plays a key role in ensuring that the team delivers.

There were also some surprising outcomes, with participants realizing the importance of efficient and rapid information sharing in a crisis that is enabled through well-connected and ICT-enabled offices. Participants also found the exercise highly stimulating, as they played the game within the device, moved around, interacted with each other, and responded to the various assessments. The learning from the ARLearn exercise was later referred to in the workshop, for example on the formation of the proof of life (POL) questions.

There were a number of technical challenges with the exercise, especially related to the problems of dependence on the wireless Internet. The live Internet enables role inter-dependencies and the ability to use manual triggers, but it requires the devices to be constantly connected to the network. The participants were also initially uneasy with the devices, and the initial challenges were more related to the usability of the devices than the game itself. As the participants got more comfortable with the devices they were able to focus on the real objectives of the exercise.

This game script was implemented in two phases. In November 2011, a dry run was organized in Budapest with staff members of the organizations. In December 2011, the actual pilot was organized in Entebbe. Here, three game runs were run at the same time, featuring three roles per run.

Although no summative and quantitative evaluation was organized for this pilot, the game organizer 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 run on personal devices, implementing a “bring your own device” (BYOD) strategy. The ability to 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 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 wireless network can sometimes result in manual triggers not arriving at a device. Automatic triggers have the advantage that they are cached on the mobile device, thus lowering the workload for the operator.
- Decreasing the dependency on Internet connection will make the game easier to export to other countries, since game organizers will not have to acquire large numbers of SIM cards or deal with wireless settings on the devices.

Conclusion and Discussion

This article presents a brief discussion of an open source and flexible architecture for mobile serious games. From an end-user point of view, creating a game is easy via a Web-based authoring environment. Players can join a game with little effort, and can download the ARLearn mobile app from Google Play. Leveraging a BYOD strategy makes organizing role-playing games—such as the UNHCR case—much cheaper to realize, as compared to acquiring specialized and expensive hardware. Furthermore, since game logic can be scripted, the cost of organizing and administering a serious game can be significantly reduced.

In the pilot organized with the employees of UNHCR, games were created with both manual and automatic triggers. For instance, in one of the first pilots, a game facilitator counted whether all participants had entered a room, and then manually triggered new content to be sent to all game participants. In the case of an automatic trigger, the dependency mechanism is responsible for making new content available. Although manual triggers offer some flexibility from the point of view of the facilitator, we have discovered that they make the game less scalable and place too large a workload on the game facilitator.

Future Work

The ARLearn tool has proved to be useful in enhancing the "realism" of UNHCR's hostage-taking simulation exercises. UNHCR's security training team plans to implement further security training scenarios in 2013–14 with the tool including but not limited to:

- Death of a staff member
- Car accident involving injuries
- Kidnapping of a staff member (terrorist, politically or financially motivated)
- Refugee threatening suicide
- Staff member injured in a mine strike that occurs with a team on a mission in a remote location
- Angry demonstration that turns into a riot outside the office
- Staff member(s) detained in a refugee camp by a group of angry refugees
- Arrest of a staff member
- Bomb blast on the street outside the office – gate destroyed and casualties among guards, but no major damage to the office itself
- Loss of the office.

Beyond security training, UNHCR also plans to use the ARLearn tool to train on scenarios in the field of emergency response, refugee protection, and program management.

One of the ongoing developments in ARLearn is the integration of displays into a serious game. Displays can add many features to a game play.

- *Content display*. Similar to broadcasting media artifacts to mobile devices, a content display makes artifacts appear on a static display. Rather than sending all players in a room an individual message, the static display can be used to show an assignment to everyone. Content display can be valuable in game synchronization points, and will also tie into the dependency system. For instance, a movie can be started as soon as all participants have checked in at the location.
- *Ambient information*. The information that is displayed relates to the environment and visualizes the game state. Showing a map of the area with the location of all players adds extra information to a player's perspective. Other options include showing the score of all players, a message board that shows the latest ten messages, and a photo pane that shows the pictures that were recently taken by players.
- *Classroom response system*. A display can also facilitate lectures to gather input from an audience. Here a typical scenario is voting. A question is shown first on the display; after voting with the mobile devices, the results show up on the display.

Currently, ARLearn provides an export/import feature for games. This provides a way of sharing games, although it is rather cumbersome. We plan to extend the metadata of both games and media artifacts. This will enable not only sharing games as whole, but also the sharing of individual smaller-grained media artifacts. Studies have revealed an inverse relationship between the size of a resource and its reusability.[7] There is thus a potential in sharing fine-grained components like a video or an audio recording. Giving mobile game authors location-based search access to these fine-grained components can speed up the creation of a serious game.

[7] Katrien Verbert, Jelena Jovanovic, Eric Duval, Dragan Gasevic, and Michael Meire, "Ontology-Based Learning Content Repurposing: The ALOCoM Framework," *International Journal on e-Learning* 5 (2006): 67–74.

Bibliography

Dede, Chris. "Immersive Interfaces for Engagement and Learning." *Science* 323 (2009): 66-9.

Gruber, Marion, Christian Glahn, Marcus Specht, and Rob Koper. "*Orchestrating Learning Using Adaptive Educational Designs in IMS Learning Design*." In *Sustaining TEL: From In-novation to Learning and Practice*. Berlin, Heidelberg: Springer Verlag, 2010.

Lave, Jeanne, and Etienne Wenger. *Situated Learning: Legitimate Peripheral Participation*. Cambridge, U.K.: Cambridge University Press, 1990.

Ternier, Stefaan, Roland Klemke, Marco Kalz, Patricia van Ulzen, and Marcus Specht. "ARLearn: Augmented Reality Meets Augmented Virtuality." *Journal of Universal Computer Science* 18 (2012): 2143-64.

Verbert, Katrien, Jelena Jovanovic, Eric Duval, Dragan Gasevic, and Michael Meire. "Ontol-ogy-Based Learning Content Repurposing: The ALOCoM Framework." *International Journal on e-Learning* 5 (2006): 67-74.

Bibliography

Delaney, Greg. "President Nazarbayev – Address to the Nation 2011 (highlights)." *Kazakh-stanLive.com* (2011).

Muzalevsky, Roman. "Key Risks Facing Central Asia in 2012 and Beyond." *Eurasia Daily Monitor, The Jamestown Foundation* 9, no. 27 (2012).

Bibliography

Burgess, Robert G.. *The Ethics of Educational Research*. London: Routledge Falmer, 1989.

Creswell, John W.. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Thousand Oaks, CA: Sage Publications, 2009.

Hodges, Jacob R.. *Mobile Learning Environment (MoLE) Project: A Global Technology Initiative*., 2013.

Israel, Mark, and Iain Hay. *Research Ethics for Social Scientists: Between Ethical Conduct and Regulatory Compliance*. London: Sage, 2006.

Resnik, David B.. *The Ethics of Science*. New York: Routledge, 1998.

Swazey, Judith P., and Stephanie J. Bird. "*Teaching and Learning Research Ethics*." In *Research Ethics: A Reader*. Hanover, NH: University Press of New England, 1997.