HAL open science

ForenCity: A Playground for Self-Motivated Learning in Computer Forensics

Frans F. Blauw, Wai Sze Leung

▶ To cite this version:

Frans F. Blauw, Wai Sze Leung. ForenCity: A Playground for Self-Motivated Learning in Computer Forensics. 11th IFIP World Conference on Information Security Education (WISE), Sep 2018, Poznan, Poland. pp.15-27, 10.1007/978-3-319-99734-6_2 . hal-02125757

HAL Id: hal-02125757

https://inria.hal.science/hal-02125757v1

Submitted on 10 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

ForenCity: A Playground for Self-Motivated Learning in Computer Forensics

Frans F. Blauw[0000−0001−5866−8335] and Wai Sze Leung[0000−0002−9015−6329]

University of Johannesburg, Johannesburg, South Africa
{fblauw,wsleung}@uj.ac.za

Abstract. Striking a balance between theory and practice in computer forensics education is considered essential to producing successful graduates with the necessary skills to take on cybersecurity challenges in the workplace. Adequately incorporating both such aspects can be particularly challenging, especially in courses or modules offered within a short time-frame. In such situations, preparing the students will require that they are incentivized to actively engage with the extensive background learning material and remain current on latest developments to correctly grasp the theoretical underpinning of the subject. In this paper, we describe the development of an adventure game to make the learning of applicable theory attractive and relevant. ForenCity takes the form of a web-based scavenger hunt in which students must apply their knowledge of computer forensics and correctly process digital evidence and progress through the game.

Keywords: Game-based learning · Self-motivated learning · Computer forensics education.

1 Introduction

A significant challenge that educators encounter when teaching computer forensics is being able to cover the subject matter adequately. Much debate exists on the topic of what content should go into a computer or digital forensics curriculum, with numerous research efforts dedicated to the topic [1, 2]. Critics of training-based courses argue that teaching the subject as a series of steps to be followed in a laboratory produces graduates lacking the theoretical underpinning of the tasks involved [2]. Similarly, academics placing too much emphasis on theoretical knowledge may result in digital investigators sorely lacking concerning practical experience [3].

As cyberthreats grow increasingly sophisticated [4], attempts to achieve a cybsersecure society may depend on educators being able to produce professionals sufficiently skilled in identifying, collecting, preserving, and analyzing digital artifacts [5]. Understanding how cybercriminals operate is essential for informing cybersecurity experts in their tasks [5].

As a branch of computer science, computer forensics naturally requires an understanding of computer systems, their underlying technology, and how these

technologies work [2]. This "prerequisite knowledge," which not all students may necessarily possess must be met as part of the curriculum. However, in cases where the offering is limited to a short time frame, covering this knowledge may be a luxury that is ill-afforded, prompting students to acquire this knowledge elsewhere – either in other modules or by engaging in self-study. Success in computer forensics can, therefore, depend on how educators find ways to encourage their students in actively taking charge of their learning.

One potential and popular strategy is to adopt a game-based approach and deliver the content so that students learn abstract concepts and explore digital forensic processes and technologies in a much more interactive manner [6]. Such an approach represents an attractive option as the learning process allows students to overcome different challenges [7].

A second approach (not unique to computer forensics), is the adoption of blended learning [8] where the best of both physical and digital worlds are combined to deliver education services and grant students greater control over their own studying [9].

The leveraging of mobile technology in several initiatives have demonstrated that the combination of both strategies above can be quite successful. In one case, a teacher was able to take their class on a virtual tour of Africa [10] while another offered users an interactive, guided tour of the less popular points of interest on campus [11].

Inspired by such cases, we propose the development of ForenCity, a mobile adventure game in which players must draw on cross-disciplinary techniques and knowledge to investigate a case. While systems for developing mobile adventures (such as ARIS) already exist, the need to develop our own arose from two shortcomings, namely (i) the need to promote variations and encourage independent problem solving amongst students, (ii) who primarily owned Android smart devices.

This paper thus reports specifically on our process of designing and developing such a game, including the game engine. Section 2 describes specific requirements for the use of games in achieving active learning, leading to details of the design of our new game engine in Section 3. Section 4 details the implementation of our game system while Section 5 describes how a particular game offering was presented using ForenCity. Section 6 discusses plans for future implementation improvements to the system while Section 7 concludes the paper.

2 Designing an Effective Learning Tool

While research is careful to acknowledge that attempts to accurately quantify the efficacy of games in learning remain immature at best, the overwhelming view regards games in a favorable light, subject to a number of guidelines [12]:

- **Promote social learning and team-teaching** – permit students to work (and learn) together.
- **Feedback that is beneficial to the attainment of success** – provide appropriate hints/clues to ensure that students can progress.

– **Balance between playability and learning outcomes** – ensure that the game is equally fun to take part in while activities align with appropriate assessment opportunities.

With these requirements in mind, the following section describes the design of the ForenCity Engine that enables us to present to our Computer Forensics students with an opportunity that encourages further self-study and to apply their theoretical knowledge practically.

3 ForenCity Engine Design

ForenCity consists of two distinct modules: the ForenCity Game Client and ForenCity Maker. ForenCity Maker is the administrative side of the ForenCity Engine to build and manage a ForenCity-based game. The player will interact with the ForenCity Game Client. A basic interface provides the player with feedback as they progress through their adventure.

3.1 Design Influence

First released in May 1984, the Adventure Game Interpreter (AGI) was a high-level game engine built by Sierra On-Line to build adventure games for series such as King's Quest, Police Quest, Space Quest, and Leisure Suit Larry [13].

Following in 1987, LucasArts developed the Script Creation Utility for Maniac Mansion (SCUMM) to ease the development of Maniac Mansion, and later the first of the Monkey Island series [14].

In 1997, Chris Jones released Adventure Game Studio (AGS) that "provides the tools to make your own adventure, for free!"[15].

All these engines (AGI, SCUMM, and AGS) use locations, characters, items, dialogue, and a basic custom scripting as building blocks to create an adventure.

For the ForenCity Engine, we took the idea of building blocks to create our adventures. Each adventure is broken up into several scenes, each with their description and requirements to complete. Individual players traverse from scene to scene to complete their adventure.

3.2 Basic Gameplay Activity

First, a game creator must create several scenes using the ForenCity Maker. Each scene will contain a basic description as well as requirements that players must meet to progress to the following scene. Only players registered on the system will be able to participate in the game. Section 4.1 discusses the ForenCity Maker in greater detail.

The player can now load the ForenCity Game Client by entering the ForenCity URL for the specific adventure. They will be presented with a login page where they can log in with credentials provided to them. Once logged in, the ForenCity Engine will see at which scene the player currently is and load the appropriate scene description.

Based on the scene description, the player can be presented with scene information, describing their whereabouts and progress. If a file is available to be downloaded, the player has access to a "Download" link. Likewise, if a YouTube video is available, the video will be loaded.

The player now has the option to "scan" the scene. When a player scans the scene, the game engine will determine their current location (based on their GPS coordinates). If the location matches the required location as set up by the creator, the engine will display a success message. Otherwise, a "nothing found" message is displayed.

If a follow-up question is required after completing a scene, the client displays the question, prompting the player to answer it. Otherwise, players will automatically progress to the following scene.

Once a player reaches the end of their adventure, they will be presented with a "success" message.

4 ForenCity Engine Implementation

ForenCity Game Engine Backend The ForenCity Game Engine (both the Game Client and Maker) is built using PHP with a JSON datastore.

PHP is a widely-used web development scripting language. Many Linux web hosts provide PHP for dynamic websites, with such packages often available on the cheapest packages.

Relational databases (such as MySQL) often come at a premium on webhosts, and so we decided to store the engine's data in JSON (JavaScript Object Notation) files. JSON is a lightweight format (as opposed to XML) and PHP has built-in JSON parsing features. Using JSON comes at the price of not having relational data to easily produce information. However, due to the linear nature of an adventure, it was not seen as a problem.

HTML5 markup was used to render client-side pages. HTML5 includes basic layout structuring but also gave us access to the device's GPS information (this is discussed later).

ForenCity Game Client The ForenCity Game Client is built as a mobile-first web application to allow cross compatibility among many devices. The only requirement is a modern smartphone with an HTML5 compatible web browser, camera, and GPS capabilities.

jQuery Mobile 1.4.5 was used to create the user interface for the Game Client. Even though newer technologies (such as Bootstrap) are available, we believed that jQuery Mobile gave us a more comfortable "all-in-one" package that immediately produces a web app that looks complete and familiar to a user.

ForenCity Maker ForenCity Maker's only requirement is a modern web browser. Bootstrap 3.3.7 was used to create the user interface that renders on both desktop and mobile devices [16]. The navigation bar and menu give easy access to all modules in the Maker.

4.1 ForenCity Maker

The Maker allows a game creator to manage an adventure using three modules: Game Manager, Scene Creator, and Player Manager.

Game Manager The Game Manager provides basic administrative functionality for the Game Engine. From this module, a game creator can enable or disable the adventure, set the adventure name, set the default GPS radius, and many more options.

Scene Creator The Scene Creator allows the game creator to create different scenes. Each scene has a *scene description* that is composed of a set of elements such as the scene's ID, its name, the description shown to the player as well as all clue data and requirements.

Scene Variables Scene variables allow the creator to personalise the adventure's experience for each player. For instance, names and adjectives can be personalised in the scene's description. The YouTube video and GPS coordinates requirement can also be set individually for each player. When a scene loads for a player, variables will be replaced by individual values as defined in the player description and set in the Player Manager.

Player Manager Each player in the game has a set of elements including their login username and password, their name how they are addressed by the game as well as their unique variables.

Player Variables For each variable that has been defined in a scene description by the creator, the actual value of the variable can be set for each player. ForenCity Maker provides a simple interface that displays all players and their respective variable definitions to allow a creator to quickly change them.

4.2 Download Implementation

Some scenes allow the user to download a file. The file can be specific to the scene or the particular player. Regardless of the file to be downloaded, the filename of the file can be set in the scene description.

Due to the stateless nature of HTTP, a download link can easily be shared between users. We overcame this by generating a unique download link for each player. The link contains a key which consisting of the player's ID and the scene for which the download is available. This key is encrypted using Aaron Francis's Urlcrypt module [17] that produces URL-friendly encrypted strings. When a player selects the download link, the Game Engine will decrypt the URL key. If it is a valid download, the file will be served to the player.

4.3 GPS Scene Requirement

An adventure relies heavily on a player's GPS coordinates. Using a browser-based web application, we made use of HTML5's geolocation component [18]. The geolocation component uses the mobile device's location services to obtain the current GPS coordinates. GPS coordinates are then loaded into form inputs and sent to the server.

Browser Requirements Apart from having location services on the device, the user must allow the site to access their location. When first loading a site that requires geolocating, the browser will prompt the user for permission. A site requesting location information must be loaded via HTTPS. Otherwise, the browser will not even prompt the user for permission.

Obtaining GPS coordinates HTML5 gives two methods to request the current location of a device.

The first is "navigator.geolocation.getCurrentPosition". This method simply asks the device for its current location. The device can then provide the GPS coordinates. However, GPS modules are normally not always running on mobile devices (in order to save power). As such, a once-off location request could potentially be quite inaccurate.

The second available method is "navigator.geolocation.watchPosition". The device is now continually polled for GPS coordinates. As the GPS modules are actively running, the accuracy will be improved over a brief period. This does use more battery power, but the requests are only fulfilled if the site is currently active.

During development, we initially made use of the first method in order to save power, but found that the accuracy was out far too often, and then opted for the second method.

Using GPS coordinates Using simple JavaScript, the GPS coordinates are loaded into two hidden form fields (latitude and longitude). This location is then submitted along with the form to the server.

The ForenCity Game Engine now received the player's current location according to their mobile device. If the requirement for the player's current scene is a GPS location, the Game Engine will determine if the player's location matches the location requirement. However, this check is not as simple as seeing if the GPS coordinates of the player directly match the GPS coordinates of the scene requirement.

GPS coordinates on consumer devices can often be off by several meters (especially if attempted indoors). The Game Engine compensates for this by first calculating the distance between the player's submitted coordinates and the scene's requirement coordinates. If this distance is within a preset maximum radius, the game engine will accept the GPS coordinates. This maximum distance can be set on an adventure-wide level, but can also be overridden on a per-scene

basis. This way, indoor scenes can have a more relaxed distance than outdoor scenes as GPS coordinates might be more inaccurate under roof or concrete.

A scene using GPS coordinates is shown as part of the ForenCity Requirements checking in Figure 1.

4.4 QR Code Scene Requirement

Some scenes need more than GPS coordinates, such as requiring the user to scan a QR code. In these cases, the player will not be presented with the option to "Scan for Clues". The scene description should give an indication that something more is required to complete the scene. QR codes can merely contain more information for the player to progress or can be used as proof that a player observed (and handled) a physical item in the real world.

Once a player scans a ForenCity QR code, the ForenCity Engine will first determine if a valid player is logged in. If not, the player will be presented with the login screen.

Since some scenes may require GPS coordinates in addition to a QR code being scanned, the player will be presented with the option to "examine" this new clue once the scene has been loaded. This allows the client to submit GPS coordinates to the server as well.

If the QR code is valid for the player's current scene, the Game Engine will perform a GPS check and allow the player to progress. If the QR code is not valid for the scene, the player will be presented with a normal "nothing found" description.

QR Code Generation The ForenCity Maker provides a QR Code generator that generates a standard URL encoded QR Code. The URL consists of the current adventure's base URL as well as a key consisting of the scene's ID in an encrypted form, once again encrypted using Urlcrypt.

4.5 Miscellaneous Functionality

Logging As ForenCity is first and foremost an implementation for a computer forensics game, it would obviously require that all actions performed by the player are logged. Such information enables us to monitor how each student progresses through the game to unlock the next part of the game. To achieve this, a log file is created that shows exactly what they are attempting. Each log entry includes the current date and time, the player's current IP address, the player's current GPS coordinates and the action that was performed:

- Player logs in and out.
- Player loads a scene.
- Player downloads a file.
- Player reaches a GPS or QR goal and progresses.
- Player is presented with and answers a follow-up question.
- Player attempts but fails, to progress to the next scene.
- Cheat attempt.

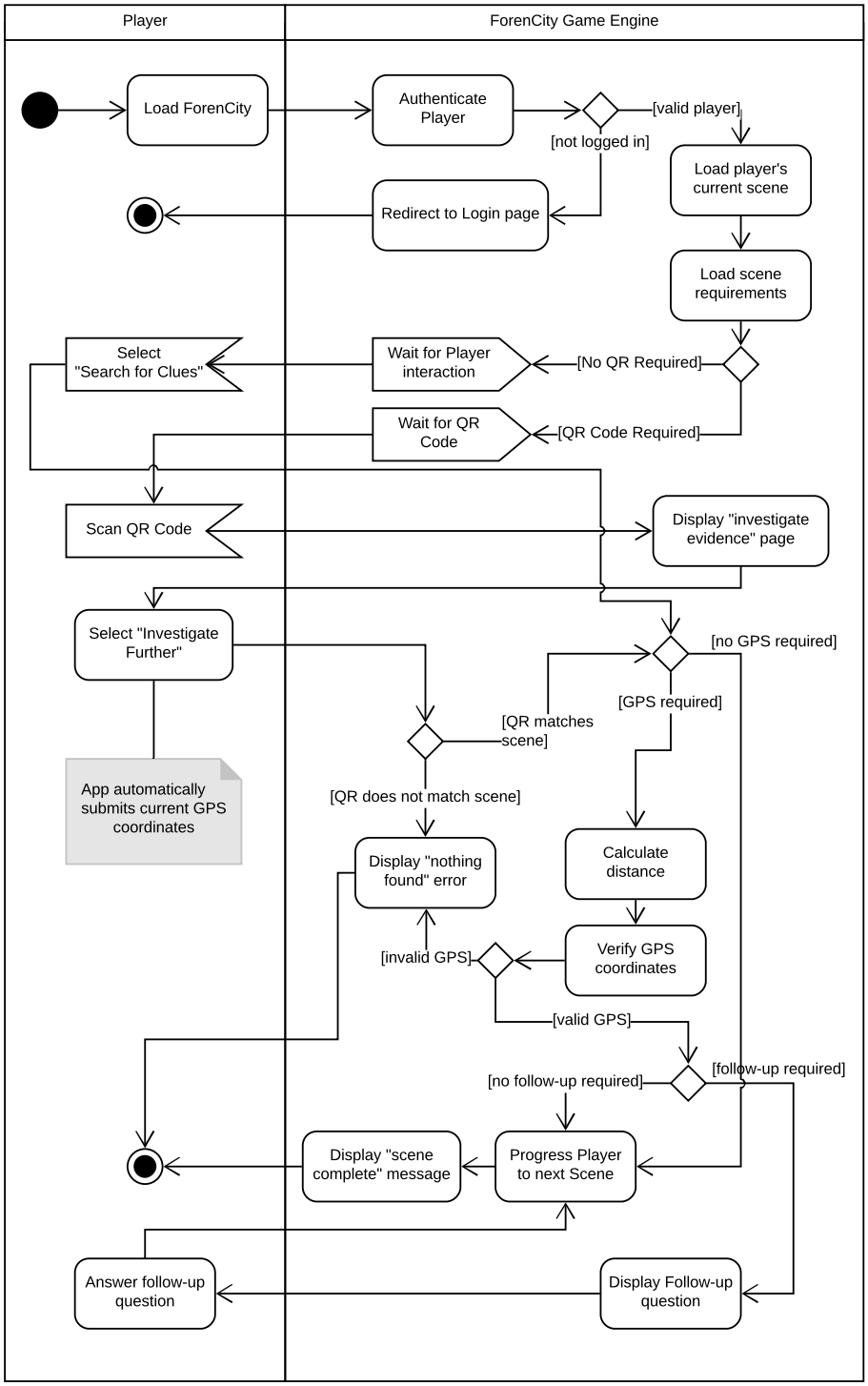

Fig. 1. ForenCity Requirement checking

Anti-Cheating Several anti-cheat features have been implemented. All cheat attempts are also logged.

- **Session Hijacking** – A player attempting to hijack another player's authenticated session will have their session automatically destroyed. This can happen if a player shares their cookie with another player. The ForenCity Engine captures the IP Address and User Agent of the player when they log in and then compares it with every page load. If either does not match, the session is destroyed.
- **Download Link Sharing** – Every download link is unique to the player (this enables each player to work with evidence files containing variations unique to that player). If the player shares the link with a fellow player, the ForenCity Engine will not allow the file to be downloaded.
- **Follow-up Question Hijacking** – A player could share the form where the follow-up question is asked with a fellow player, in order to potentially bypass an actual GPS or QR code check. Every Follow-up Question form has a unique key for every player. If the key is not present for the correct player, the answer will be rejected.

Having described the various features of ForenCity, the following section will now describe a particular case set up for a group of students.

5 Solving for a Kidnapping

In 2017, we "deputized" 15 students as investigators, tasking them to investigate the kidnapping of a mining magnate's young daughter. The case is, in essence, a race against time (although students were given the space of a week to conclude drafting a final report on how they reached the end goal).

This section showcases select scenes from the game, detailing the activities taking place, and expected outcomes from the students.

- **Scene 1: Police Headquarters (Debriefing)**
  - **Description:** Investigators view a video in which Detective Sergeant Tango debriefs them on their latest case: the kidnapping of Gugu, the daughter of a mining magnate. The only clue available at present is an email that the perpetrator(s) sent to Gugu's father from a disposable temporary email address. Attached is a photograph of Gugu along with the demand for 888 Bitcoin, to be credited into a Bitcoin wallet.
  - **Expectations:** The student must analyze the email for clues. The student should notice (in their final report) that the email is from a throw-away address. The attached photograph, however, contains metadata that includes coordinates suggesting Gugu's last known location (when the photograph was taken). If the student is not familiar with how GPS coordinates work, they will now need to research how to decipher the values to visit the location physically.

- **Scene 2: Gotham Heights (The First Witness)**
  - **Description:** Investigators arriving at the location of the coordinates will be greeted by a cranky caretaker who complains about a âĂIJsuspicious individual with a crying kid.âĂİ Being very alert, he not only gives the investigator a description of the van that the suspect drives (the name of a business is given) , but also provides a USB that the suspicious individual had dropped in a hurry.
  - **Expectations:** To "receive" the USB, the student downloads a file of the image. A cursory scan of the USB reveals a single file, the same photograph of Gugu attached in the ransom email. However, by making use of appropriate forensic tools, the student will recover a deleted file: a password-protected PDF. Students may attempt to crack the password with appropriate tools. However, it is expected that the student should rather pursue the other clue about the van as the business refers to a popularly known business that can be physically found on the university campus.
- ***«Scenes 3 and 4 cut for brevity»***
- **Scene 5: The Bulgarian Consultant**
  - **Description:** The investigator makes their way up the building, arriving at the fourth floor where they are greeted by the Bulgarian Information Broker who is rumored to have many connections and holds the right answers. She gestures towards a code on her office window before returning to her work.
  - **Expectations:** Being in a building where it will be difficult to ascertain the student's position in terms of floor level, a QR code is provided as the next clue. We had previously made arrangements with our faculty's librarian to play the role of the "Bulgarian Information Broker". The student scans this QR code to obtain another set of coordinates that will lead them to an Internet café.
- ***«Scenes 6 and 7 cut for brevity»***
- **Scene 8: The Locker**
  - **Description:** After questioning the witness in the previous scene, the investigator approaches the locker where the suspect is believed to have been loitering about previously. Some answers have been uncovered but there may be further clues that could cement the case for the investigator. Unfortunately, the locker is secured with a padlock and they are only in possession of the first three digits of the four-digit combination. It is also now, that Detective Sergeant Tango calls, ordering the investigator to finalize their investigator and put together a report.
  - **Expectations:** The student may attempt to open the locker in one of two ways: since only one digit is missing, it is possible to apply brute force and test out all ten possible combinations. Alternatively, a physical clue in the previous step may yield indented writing of the combination written on a notepad. The game is set such that a variety of choices made throughout the game may yield the same end result. However, the majority of marks awarded will come from the student's ability to

correctly motivate their rationale for carrying out a certain action, as detailed in their final report.

6 Future Implementation

Following with our influences (Section 3.1), we want ForenCity Engine to have more than only scenes. As indicated in Section 2, we plan to create a more non-linear adventure where players will have the opportunity to explore all scenes at any time and interact with characters, interactive objects, and an inventory system.

- **Non-Linear Gameplay** – the current ForenCity Game Engine only allows for a linear story to be told. However, in real-life, different investigators' reasoning might lead them to different clues first. As such, we want most scenes to be available from the start of the adventure, but giving (or removing) information as different tasks are completed. Different scenes might also change depending on the time of day.
- **Characters** – Forensic investigation is not only limited to inanimate objects. Often interaction (and interrogation) with other people will be required. We want to develop a character component that will include a complete dialogue tree system. Players will have the opportunity to meet with these virtual characters and have interactions with them. Depending on their dialogue choices, characters will either reveal or withhold vital information.
- **Interactive Objects** – Forensic investigate will require interaction with objects at a scene. Similar to Characters described above, we want to implement a component where the player can interact with virtual objects to reveal more information. Interactions can be limited to something simple such as switching a "light" on or off. More complicated interactions can include performing a search and finding files on a computer.
- **Inventory** – Often one piece of evidence will lead to more and different clues. As the player progresses through their adventure, they should be able to pick up items along the way. Items can then be used alongside other items, objects or even characters.

7 Conclusion

As seen in our discussion on the various features of ForenCity, we have developed a platform that enables educators to create engaging and customized problem-based assessments in the form of an adventure game that takes students beyond the physical classroom. Based on participants' feedback, we enjoyed a rather positive and enthusiastic response from students excited to draw on their computer science and computer forensics knowledge in order to unlock the next clue and reveal how the next chapter in the mystery would unfold.

In ForenCity, we were able to create an environment that augmented physical items and spaces on our institution's campus with virtual characters and props to

guide each student through a police investigation that developed as they engaged with both the physical and virtual props around them. Restricted only by our imagination and storytelling skills, ForenCity could potentially be used to assess students on a variety of other cybersecurity skills. Beyond implementing the extra features in ForenCity, we look forward to expanding ForenCity's storyline, inviting students in other subjects to test their mettle while getting to apply and experience the skills and knowledge they have acquired thus far.

References

1. Irons, A.D., Stephens, P., Ferguson, R.I.: Digital Investigation as a distinct discipline: A pedagogic perspective. Digital Investigation **6**, pp.82–90 (2009).
2. Lang, A., Bashir, M., Campbell, R., Destefano, L.: Developing a new digital forensics curriculum. Digital Investigation **11**, pp. S76–S84 (2014).
3. Willems, C., Meinel, C.: Online assessment for hands-on cyber security training in a virtual lab. In: Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON), pp. 1–10. IEEE, Princeton, NJ, USA (2012).
4. Six Cyber Threats to Really Worry About in 2018, https://www.technologyreview.com/s/609641/six-cyber-threats-to-really-worry-about-in-2018/.
5. Cyber Security Awareness Month: Cyber Security vs. Cyber Forensics, http://www.stevenson.edu/online/blog-news-events/cyber-security-vs-cyber-forensics.
6. Pan, Y., Schwartz, D., Mishra, S.: Gamified digital forensics course modules for undergraduates. In: 2015 IEEE Integrated STEM Education Conference, pp. 100–105. IEEE, Princeton, NJ, USA (2015)
7. What is Game-based learning? https://www.game-learn.com/what-is-game-based-learning/.
8. Blended learning is the future, https://mg.co.za/article/2017-03-17-00-blended-learning-is-the-future.
9. Graham, C.R.: Blended Learning Systems: Definition, Current Trends, and Future Directions. In: Handbook of blended learning: Global perspectives, local designs, pp. 3–21. Pfeiffer Publishing, San Francisco, CA, USA (2006)
10. ARIS: A Field Day Lab Experiment, https://fielddaylab.org/make/aris/.
11. Garay-Cortes, J., Uribe-Quevedo, A.: Location-based augmented reality game to engage students in discovering institutional landmarks. In: 2016 7th International Conference on Information, Intelligence, Systems Applications (IISA), pp. 1–4. IEEE, Princeton, NJ, USA (2016)
12. de Freitas, S.: Are Games Effective Learning Tools? A Review of Educational Games. Educational Technology & Society, **21**(2), pp.74–84 (2018).
13. Adventure Game Interpreter, https://en.wikipedia.org/wiki/Adventure_Game_Interpreter.
14. Script Creation Utility for Maniac Mansion (SCUMM), https://en.wikipedia.org/wiki/SCUMM.
15. Adventure Game Studio, http://www.adventuregamestudio.co.uk/.
16. Bootstrap, https://getbootstrap.com/.
17. Urlcrypt, https://github.com/cheerful/URLcrypt.
18. HTML5 Geolocation, https://www.w3schools.com/Html/html5_geolocation.asp.