Development and Initial User Evaluation of a Virtual Crime Scene Simulator Including Digital Evidence

Alleyn Conway[1], Joshua I. James[2], and Pavel Gladyshev[3(✉)]

[1] IT Planning, an Garda Síochána, Phoenix Park, Dublin 8, Ireland
alleyn.conway@ucdconnect.ie
[2] DigitalFIRE, Hallym University, Chuncheon, South Korea
joshua@cybercrimetech.com
[3] DigitalFIRE, University College Dublin, Dublin 4, Ireland
pavel.gladyshev@ucd.ie

Abstract. Imagine the following scenario: an inexperienced law enforcement officer enters a crime scene and – on finding a USB key on a potential suspect – inserts it into a nearby Windows desktop computer hoping to find some information which may help an ongoing investigation. The desktop crashes and all data on the USB key and on the Windows desktop has now been potentially compromised. However, the law enforcement officer in question is using a Virtual Crime Scene Simulator and has just learned a valuable lesson. This paper discusses the development and initial user evaluation of a Virtual Crime Scene Simulator that includes the ability to interact with and perform live triage of commonly-found digital devices. Based on our experience of teaching digital evidence handling, we aimed to create a realistic virtual environment that integrates many different aspects of the digital and physical crime scene processing, such as physical search activities, triage of digital devices, note taking and form filling, interaction with suspects at the scene, as well as search team training.

Keywords: Virtual crime scene simulator · Digital forensic training · Unity3D · Digital evidence

1 Introduction

As with any discipline, the opportunity to practice, learn and apply such learning to a real-life scenario instills confidence within oneself. It allows mistakes to be made, actions to be evaluated and tasks to be set. In the sometimes high-pressure environment of a crime scene search involving complex physical and digital evidence, one incorrect action can be detrimental to any future judicial proceedings.

There have been a number of projects that attempt to virtualise crime scenes either for entertainment, educational or, more recently, for law enforcement training purposes. The examples of entertainment involving crime scene simulations include the popular Facebook game 'Criminal Case' [1] and the Interview Simulator game [2]. The Interview Simulator is a relatively simple, static interaction that attempts to simulate interviews in the

context of police investigation. While such entertainment represents some educational value for law enforcement, it does not accurately represent real crime scenes.

The F.B.I. realised the need to accurately represent real crime scenes, and began working with Epic Games' Unreal Engine 3 [3] in an effort to create realistic, complex 3D simulations of crime scene activities in a first-person point of view. Such an effort is similar to the 3D game released by the U.S. Army, called 'America's Army' [4]. Other simulator games specifically for law enforcement include firearms and other tactical training [5].

Although existing specialised simulators are valuable as educational tools, there are problems hindering their practical usage for digital forensics training. First and foremost, existing crime scene simulators do not normally represent digital evidence in the crime scene but rather cover crime scene reconstruction and investigation, e.g., Crime Scene Virtual Tour [6]. Also, these systems do not easily allow for the modification of scenarios that can match the training course. Overcoming these challenges, while preserving the realism and complexity of the simulated experience has been the focus of the work described in this paper.

1.1 Contribution

This work contributes to the field of digital forensic investigation by providing considerations for crime scene investigation training that includes digital evidence; providing a 3D virtual environment in which law enforcement can practice learned crime scene search and seizure procedure; and evaluating such systems with law enforcement officers during classroom training.

2 3D Technology and Crime Scene Simulation

With the advent of software such as Unity 3D, a powerful cross-platform 3D engine and Virtual Reality (VR) headsets such as Oculus Rift a developer now has access to technology that only a few years ago would have been the preserve of a Hollywood film studio.

There are multiple examples of 3D technology been employed by law enforcement, military and educational institutes. Whilst all have varying end users the methods the applications employ allow for effective knowledge transfer.

Crime Scene Virtual Tour (CSVT) [6] allows the user to reconstruct a 3D crime scene using photos, provides a 3D measurement feature and thereafter walks through the scene. By uploading photos of the crime scene the user is able to zoom, pan, tilt and rotate the scene and relive the crime scene. Unlike the Virtual Crime Scene Simulator developed as part of this project, however, it does not offer the ability to interact with devices found at the scene. Another solution is offered by AI2-3D [7], a Canadian company, specialising in the reconstruction of crime scenes using 3D forensic visualisation for court presentations.

Beginning in 1960 "the armed forces took the lead in financing, sponsoring, and inventing the specific technology used in video games" [8]. Spacewar!, which some

historians would consider the first video game, was developed by graduate students at MIT who were funded by the Pentagon. Forward 50 years and we now see sophisticated First Person Shooter games such as American's Army or Full Spectrum Warrior being played by gamers but also being used by military forces for training [9].

3   Development of the Virtual Crime Scene Simulator

There are a number of reasons as to why a new Virtual Crime Scene Simulator is needed for digital forensic investigation training. First, crime scene processing is a vital part of investigator training that can only be learned through practice. The initial aim of the project described in this paper was to create a virtual environment that would allow distance-learning students to familiarize with and practice the key elements of the crime scene search before a trip to a dedicated training facility for hands-on training.

In comparison with physical crime scene simulations, virtual environments have a number of distinct advantages:

- virtual crime scenes are less expensive and faster to setup;
- virtual crime scenes could be big, complex, and/or highly unusual;
- virtual crime scenes can be used to perform joint training sessions where team members are geographically far apart;
- virtual crime scene could allow straightforward simulation of live triage and crime scene processing using virtual machines.

These reasons provide the motivation for development of the Virtual Crime Scene Simulator, and minimum final functionality for the system.

Unity3D was chosen as the base platform for the Virtual Crime Scene Simulator because it comes with a free feature-complete development environment. Another advantage is support for multiple platform development. Applications can be created for iOS, MACs, PCs, Steam, Xbox and others.

The Virtual Crime Scene Simulator provides a realistic crime scene with configurable search options. In addition to physical crime scene simulation it encapsulates virtual machines, imitating digital devices in a crime scene along with an Artificial Intelligence Markup Language (AIML) based chatbot [10] providing interactivity with an avatar.

Before commencing development, a 3D representation of a house was created containing three rooms, a hallway, a hidden room and outside scenery. The creation of such scenery is a straightforward application of Unity3D and can be created without any development experience whatsoever. Following the creation of the scenery the next step was the creation of game objects. These set the scene for the user and included items such as:

- Furniture
- Props
- Hardware devices, e.g., mobile phone, desktop, hidden laptops
- An avatar for interrogating

The intelligence and internal logic of the game objects was written mostly in Javascript (Unityscript), which is built into Unity 3D engine. The following gaming actions were developed and applied to the game objects:

- Issuing of a warrant before a search could commence
- Physical searching of game objects
- Labeling, exhibiting of possible evidential item
- Dismantling, examining and seizure of game objects
- Note taking during crime scene searching
- Interactive avatar who can be treated as a suspect and/or witness and questioned appropriately
- Live examination of digital devices such as mobile phones, desktops, laptops, Smart TVs using multiple options
- Live feedback with reference to ACPO principles [11] depending on device interaction

The Virtual Crime Scene Simulator is an open source project. The source code and the binary installers can be downloaded from the project website [12].

If a trainer needs to amend the scene then he or she will need to be familiar with the scene functionality in Unity3D along with knowledge of how the scripts in the application interact with each other. Should the trainer need to add logic over and above what the application does then the existing Javascript can be amended or new code added. Once this is achieved the trainer has the ability to create additional devices or delete existing ones, add new rooms, create new tasks, etc.

Figure 1 shows the initial screen of the current version of Virtual Crime Scene Simulator. Upon installation the user can select various setup options via an administration menu. These options allow the user to

- turn on/off background noise
- turn on/off virtual integration with external devices
- detail what OS should be used on simulated digital devices – there is a separate configuration option per each simulated device.
- choose type of pre-programme search scenario – urgent or routine.

3.1   Sample Walk-Through

Upon entering the crime scene the user is presented with a number of tasks she or he is required to perform. Currently, these include

- Exhibit at least 3 items
- Discover a hidden object
- Begin an interrogation
- Get suspect's date of birth
- Save log files

Once the simulator commences, the user has the ability to walk around all rooms within the scene but can only start interacting with the game objects once a warrant has been presented to the avatar. Once done, a suite of options are now available.
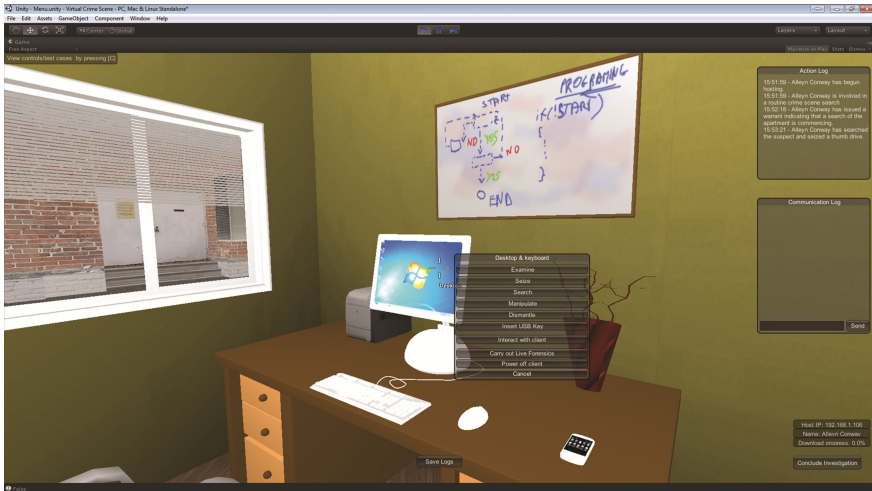
Fig. 1. Opening presented to the trainee when starting the virtual crime scene simulation in single-user mode.

Fig. 2. Virtual environment shown to the trainee when choosing to analyze the suspect computer in the virtual crime scene simulator.

If the user wishes to interact with a digital device an option "Interact with <device>" is available in the list of actions for each digital device. Currently device interaction is fully realized only on the Microsoft Windows version of the Virtual Crime Scene Simulator. It is implemented using Virtual Machines running under VirtualBox. During the device interaction phase, the user is shown the console of the virtual machine in a full-screen mode (see Fig. 2). The user may now interact with the VM and upon finishing return to the crime scene. Through the Admin menu the trainer can elect which operating system (either Microsoft Windows or Ubuntu Linux) particular devices will be running when called, e.g., the laptop could be running Ubuntu whilst the desktop could be running Windows.

Fig. 3. A list of action options that is available when interacting with the suspect desktop on-scene.

Virtualbox VMs are called from the simulator using Windows shell scripts (.BAT files), which can be adapted to use a different virtualisation manager. Nircmd utility [13] was also utilised to ensure that the virtual client is always displayed on top of the Unity backdrop and is centered appropriately.

When the user interaction with the virtual device is finished the virtual client is rolled back to its original snapshot and the user is returned to the crime scene. The snapshot rollback ensures that the trainer does not need to restore all the virtual clients back to their original state for the next training session

In addition to interacting with digital devices there is also the ability to search for, manipulate and examine game objects, e.g., moving a bookcase reveals a hidden room or examining a bed reveals a laptop under the covers (see Fig. 3 as an example).

As well as interacting with inanimate and digital objects, the user has the option to interrogate an avatar. The avatar's intelligence is simulated using a back-end chatbot application. The chatbot responds to questions which are pre-programmed using Artificial Intelligence Markup Language (AIML). The user can choose whether to question the avatar as either a suspect or a witness. All questions and responses are logged in a time-stamped log file.

Upon completion of the user's session the application saves the following log files:

- Action log detailing all options selected by the user and feedback from the application
- Notes log detailing any notes taken by the user
- Interrogation log detailing all questions and responses from the avatar
- Communications log detailing all chat which took place with other search team members

These log files can then be analysed by a trainer and used to give detailed feedback to the individual students and the team as a whole.

4   Evaluation of the Virtual Crime Scene Simulator

In August 2014 the opportunity arose to have the application tested by members of Law Enforcement from a developing country. These law enforcement officers were in the process of receiving a 5 month in-house training course on digital forensic and cyber-crime investigations. Such training included a crime scene investigation component.

In the current version of the Virtual Crime Scene Simulator, virtual machine integration executes on a Windows OS only, so the ability to launch a VM was excluded from the training exercise. However, all other device interaction functionality, such as switching devices on and off and connecting USB sticks to it, was available.

Overall, 6 participants used the Virtual Crime Scene Simulator. Each had various levels of experience with computers in general and with computer games.

Before training, the trainees had at least 3 years experience as criminal investigators. Three out of the group had some experience with digital devices, but considered themselves 'beginner' level. The remaining trainees had no experience with digital investigations.

After lectures on related topics, the trainees were instructed to install the Virtual Crime Scene Simulator (VCSS) on their laboratory workstations. Each trainee was able to download, install and run the VCSS with no issues.

As mentioned, each of the trainees were experienced law enforcement officers with very different computing backgrounds. This was immediately apparent once the VCSS was started. Those with little computing experience, especially those that claimed to have never played video games before, found it difficult to interact with the VCSS. The trainees with computer gaming experience were able to use the system with no issues. Trainees with little-to-no experience took approximately 30 min to get familiar with the controls.

The exercise goal was to complete the previously mentioned pre-defined tasks. The system provided an automatic checklist of tasks, and indicated whether they have been completed or not. The trainees were instructed to let the teacher know when they have finished.

Like a normal crime scene, there are a number of objects in the virtual crime scene that may be related, but are not explicitly listed as objectives. The trainees could choose whether to seize, analyse, deconstruct or leave any other objects they found.

After each trainee finished, we collected Virtual Crime Scene Simulator log files detailing each trainee's search and seizure activity, and had each student fill out a survey relating to their experience.

4.1   Virtual Crime Scene Simulator Questions

This section covers the virtual crime scene simulator survey results from the trainees immediately after using the VCSS followed by a brief analysis of the results.
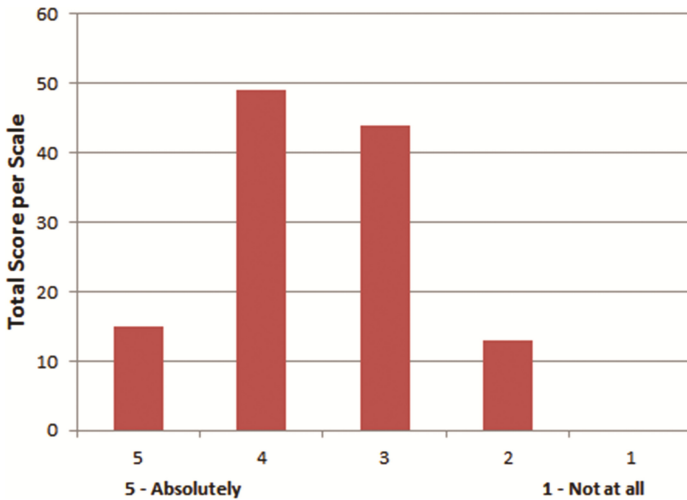
Fig. 4. Allocation of scores combined from 20 questions from all respondents.

Twenty questions were posed to the trainees, which covered the following areas:

- User Interaction
- Educational Value
- Technical aspect of the application

The trainees were asked to assess each question based on a scale of 5 (Absolutely) to 1 (Not at all).

In addition to these questions the trainees were also asked to (1) detail what modifications they would like to see added to the application, (2) what functionality did they particularly like and (3) to document issues which they had with the application (Fig. 4).

As can be seen from the above graph the application achieves a high score in the 4 to 5 scale. The fact that some of the participants were never used to play FPS (First Person Shooter) games needs to be taken into account when viewing the 2–3 scale totals (Figs. 5, 6 and 7).

4.2 Survey Analysis

The use of Linux version for the evaluation highlighted a number of bugs which did not exist on the Windows OS version of the VCSS and is an area which needs further improvement for future releases.

One student, who had computer experience and had used a FPS game before, scored the application high. It is possible that the average scoring attributed by the non-computer experienced users related more to their lack of experience with computers and FPS games. Possibly, with more experience such trainees would focus on the scene search rather than concentrating on manipulating the game for the first time.

The results in Sect. 4.1, on average, score highly amongst the participants in relation to the educational purpose of the application and indicate that there are enough tasks to
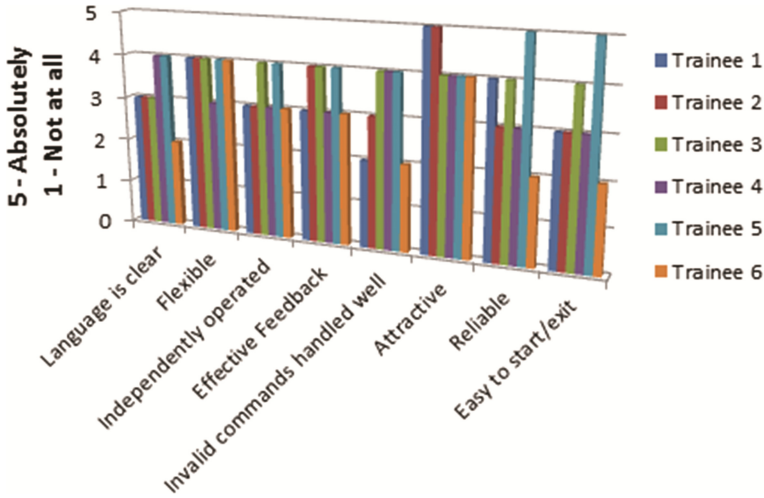
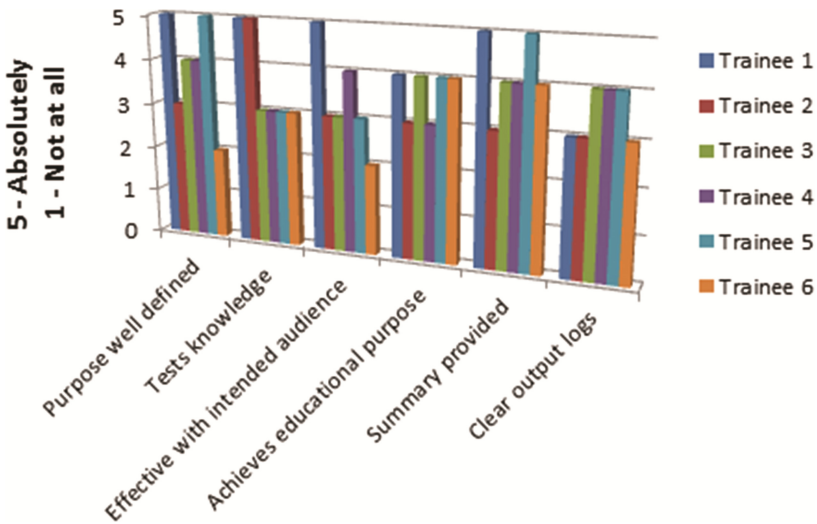**Fig. 5.** Responses relating to user interaction.

Fig. 6. Responses relating to the educational value of the application

motivate the users to continue using the program. This is very encouraging given the usability bugs within the Linux version of the application.

It is worth noting that one participant marked the application with a mark of 2 in 9 questions but yet scored it 4 when it comes to stating that the program achieves its purpose from an educational viewpoint.
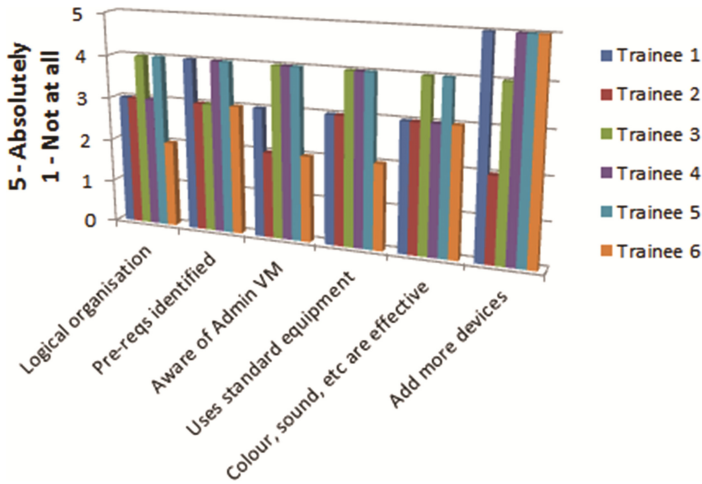
Fig. 7. Responses relating to the technical aspect of the application.

5   Trainer Comments

From a training perspective, the VCSS was very useful for reinforcing the theoretical concepts presented to students prior to the exercise. No students had experience with digital evidence seizure in a crime scene, and their reactions to the VCSS were very much like what we have seen with inexperienced investigators in the field. Specifically, rushing straight to digital devices, non-structured searching, not adhering to ACPO or similar guidelines, etc.

From a trainer's perspective, the logs that are produced by the VCSS provide a clear indication of what the trainee was trying to achieve, as well as the crime scene search process (or lack thereof). This information proved to be extremely valuable for identifying areas of confusion, and was used both to discuss such concepts again, and to better illustrate how such concepts would be conducted in a real crime scene.

6   Conclusions and Future Work

The application, even when used by participants with little prior computer experience, offered a valuable insight into real crime scene processing with digital devices, and the options faced by an investigator upon entering a possible crime scene. Notwithstanding the Linux-specific issues which arose and the lack of computer experience amongst the participants it was encouraging to note that users still got benefit from the application.

As with any application there is always room both for improvement and expansion. It is now planned to incorporate functionality to make the application more versatile which will include, though not limited to, the following:

- Creation of independent game objects so they can be placed into any Unity Scene for re-use

- Customize setup using configuration files, e.g., localisation of narrative, task creation, scenario types, etc.
- Extending interaction with virtual environments for Linux and OSX operating systems
- Incorporating Oculus Rift plug-in
- Record and playback for further training purpose.

Further enhancements and plug-ins are planned for further releases to ensure that this application over time will become a best-in-practice learning application for crime scene investigators.

References

1. Criminal Case Game. https://www.facebook.com/CriminalCaseGame (2014)
2. Interview Simulation. http://www.horton.com/portfoliointerview.htm (2013)
3. Makuch, E.: FBI using Unreal Engine 3 crime scene sim. http://www.gamespot.com/articles/fbi-using-unreal-engine-3-crime-scene-sim/1100-6368256/ (2012)
4. Osborne, S.: Americas Army. http://www.gamespot.com/reviews/americas-army-operations-review/1900-2895424/ (2002)
5. Lee, C.: Police simulator training more than just a video game. http://www.dailyillini.com/news/campus/crime/article_1cdc97e1-3d6a-5df4-9fdc-1f394c2a8d44.html?mode=story (2013)
6. Crime Scene Virtual Tour. http://www.crime-scene-vr.com/ (2011)
7. AI2-3D. Forenisc Mapping, Analysis & Visualization. http://ai2-3d.com/ (2014)
8. Mead, C.: War Play: Video Games and the Future of Armed Conflict. Eamon Dolan/Houghton Mifflin Harcourt, Boston (2013)
9. Perez, M.: Military Advertising and Simulation Training: U.S. Army's Use of Video Games. http://www.academia.edu/6823477/Military_Advertising_and_Simulation_Training_U.S._Armys_Use_of_Video_Games (2012)
10. Shawar, B.A.: A Corpus Based Approach to Generalising a Chatbot System: Applying Simple Natural Language Processing Techniques to Build Knowledge Base of ALICE Chatbot System LAP Lambert Academic Publishing. dl.acm.org/citation.cfm?id=2132757 (2011)
11. Association of Chief Police Officers. ACPO Good Practice Guide for Digital Evidence. http://www.acpo.police.uk/documents/crime/2011/201110-cba-digital-evidence-v5.pdf (2012)
12. Open Source Digital Forensics Reference Site. www2.opensourceforensics.org (2014)
13. NirSoft. NirCmd v2.75. http://www.nirsoft.net/utils/nircmd.html (2013)