Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2012) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

Virtual Worlds for Serious Applications (VS-GAMES'12)

Serious game in security: A solution for security trainees

Vítor Oliveira[a,c], António Coelho[a,b,c], Rui Guimarães[c] and Carlos Rebelo[c]

[a]Departamento de Engenharia Informática, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal
[b]INESC TEC (formerly INESC Porto), Porto, Portugal
[c]3Decide, Palcos da Realidade, Porto, Portugal

Abstract

Serious games have been used with success for training field operatives in tasks where there is a danger of injury or life threatening situations. This paper presents the development of a serious game aimed at the areas of security and safety, supporting the training of specialists through supervised situational scenarios. The training plans involve security against third parties, focusing on social level security at a corporate level, and also safety actions on events such as floods and fires in buildings/facilities. The game provides a 3D virtual environment of the real location/facility to be secured and a multiplayer platform to allow collaborative training and supervising.

Keywords: Serious Game; Security; Safety; Multiplayer; Cooperation

1. Introduction

The area of Security is focused on the protection of people and goods against damage, loss or crime. Security teams are trained to take into account the actions of people attempting to cause destruction and to respond effectively by taking preventive actions.

Serious games allow users to be involved in a motivating environment and, taking into account the amount of available tools, it is possible to integrate realism and simulation components, promoting learning and the effective acquisition of practical skills. Serious games can thus be designed and developed to provide a virtual platform that offers a realistic environment suitable for the needs of training plans in the area of Security.

This work has been conducted with the collaboration of a private security company that provided knowledge on the training plans of its professionals. From informal interviews with experts from the company, three main topics were selected: Intrusion, Fire and Flood. It is also relevant that different locations involve different approaches: security and safety of an office building differs clearly from a football stadium during a match. In this context, the concept of "situated learning" arises, promoting learning in contexts that reflect how skills will be used in real-life situations. It also emphasizes social collaboration in order to solve problems through dialogue and discussion.

On-site simulations that recreate problematic situations to test the reliability of security plans are often required for their approval. These exercises that are performed in the real location and with real people may have a high economic impact. Serious games promote learning through practice in a safe, but also realistic, environment. Nevertheless, existing solutions (section 2) for the area of Security are rare and do not provide the necessary realism or are not flexible enough for this approach. Being Security an area with a strong collaborative factor, it is important to have a component directed to the same goal. However, in existing solutions, the collaborative component falls to the background plans through artificial intelligence or exchange of ideas. The solution proposed in this paper fills this gap, allowing cooperation between players in real time.

2. Related Work

In order to determine the best practices in the areas of Security and Safety, a study on serious games solutions was conducted, analyzing the degree of adequacy to the problem under consideration.
"RescueSim" is a multi-agency training software based on a virtual environment that prepares public safety professionals to real-life incidents. It allows teams to gain experience in emergency response to incidents allowing the teams to determine the best strategic response by observing the consequences of their decisions [1]. Sponsored by NATO, "Boarders Ahoy!" is a multiplayer game offering a 3D environment for training boarding teams to perform maritime interdiction of suspect vessels. The players are able to interact with the crew but the dialog is controlled by a limited set of options for directing the training [2]. In "Die Polizei Simulator 2011", the player assumes the role of a police officer who has to respond to several cases of theft, riot control, etc. In "Agent Surefire", the player must identify the points in which the company is vulnerable to theft and leakage of confidential information [3]. In "Versailles 1685" the player is a guard that must prevent the destruction of Versailles, by searching for a bomb and the person who would activate it [4].

2.1. Important Features

"RescueSim" offers features of cooperation and the possibility of editing the scenarios in order to be more convenient for learners. This flexibility is important to train teams of professionals on specific scenarios. "Boarders Ahoy!" features interactive characters and objects that provide only a limited set of interactions. This feature helps to direct the player on the training plan, avoiding too much dispersion. "Die Polizei Simulator 2011" has also the possibility of interacting with NPC (Non-Player Characters) and the training is performed over a sequence of actions. "Agent Surefire" allows the player to control its avatar as in an adventure game. Although it is suitable for learning procedures, the mechanics does not depend on real time events, and the response time cannot be included in the training plans. "Versailles 1685" has a great emphasis on the dialogue between characters but conversations are predetermined. For a multiplayer game it is not the best option since it should adapt to the unpredictability of the behavior of the players.

2.2. Limitations

"RescueSim" does not allow scenario editing in real-time limiting the options on the training supervisors. "Boarders Ahoy!" lacks event randomness. The other games lack multiplayer feature, so there isn't much of a cooperation component. "Agent Surefire" and "Versailles 1685" can be memorized if a replay is done, making them unsuitable for recurrent training plans.

3. Game concept

The purpose of this game is to provide an engaging solution for training security professionals on specific scenarios. The game is based on a realistic virtual environment, enrolling multiple trainees at an operational point of view, under the control of a supervisor. The main purpose of the supervisor is to scatter cameras and other equipment through a scene to help the players and simulate problematic situations. The supervisor can trigger hazards such as fires or floods to provide challenges for the players. It's up to the security team to watch the cameras, make patrols and take actions in response to these problems. Other problems can be also generated by the NPC that represent regular people on the scene. They can start yelling or fighting, and it's up to the security team to find the best way to deal with these situations. To harden things, the players can also play as regular people, providing more randomness to the training plan. With this scheme in mind, the game can be adapted to any situation at any location.

The game was conceived as a multiplayer game in order to provide interaction between trainees and supervisors, both in cooperative and competitive modes. The player can take several roles: Security, Civilian or God. Each has its own function in the game, even though Security is the role most related to the trainee. The view is in first person in order to implement a more realistic and emotional perspective. The virtual environment is populated with NPC that perform the role of Civilians. These NPCs can interact among themselves and also can be approached by the players in order to interrogate and direct their behavior. To monitor and supervise the virtual environment, there are cameras that can be viewed on displays positioned in a wall in the control center. The positioning of cameras, and other objects like the fire extinguishers, depends on a previously selected plan containing their positions and attributes. At the end of the game the trainees are faced with a score on their individual performance and as team. The individual score is only shown to players of type 'Security' and is complemented with feedback about the performance of their individual actions.

The players are challenged by events that could compromise the security and safety of the place, which are randomly generated or supervised by the God player. Players can cooperate to solve these problems faster to obtain a better score. Following the experts' opinion, the game comprehends three main security and safety issues: Intrusion, Fire and Flood. These were addressed as the main effectors on the scoring system. Score is obtained by making the correct choices of actions in response to the problems and the faster the response the better the score.

Before starting to plan the game itself, there are a number of base factors to consider. These involve pedagogical as well as motivational factors.

3.1. Pedagogical factors

This game focus on the operational level and at the end of the training plan the trainees area expected to be able to solve security and safety problems following the correct procedures and at the proper response time. Although most companies follow the same protocols, it is desirable to know which are the ones that the company complies. In that sense, it is convenient to make an interview with a member of the company in order to know the requirements to have a successful training plan according to the company's procedures.

Knowledge about the location is a key element in creating a serious game in this area, both the physical site and the people that populate it. Doing surveillance on a train station is different from doing surveillance in an office. The existing protocols for the field of private security, 'International Code of Conduct' [5] and 'Code of conduct and ethics for the private security sector' [6], don't differ substantially from each other and serve as a starting point to understand the key concepts in this area.

Being a serious game there must be an assessment component to monitor how the trainee progresses on the training plan. This is done through a scoring system and direct feedback directs the learning process.

3.2. Motivational factors

The use of a 3D game engine is ideal for this application as it provides a high level of realism and immersion and it is possible to provide simple solutions at the functional requirements level, but at the same time allowing some complexity in the non-functional requirements.

The scoring mechanism, associated to the multiplayer mode complement the motivational component, given that 'online gaming' is primarily driven by social reasons with players seeking cooperation and competition [7].

For this, there must be an implementation of a network architecture that enables players to communicate with each other. In the case of Unity3D, this architecture can be made through a client-server architecture. The architecture adopted in the prototype is described in Fig. 1 and can be adapted to any game in this area that has a multiplayer component.

It is expected that the target players do not play often at home. It is therefore necessary to simplify most of the controls and concepts, so that the complexity of the game is transparent to the players. Keyboard movement and interaction through mouse should be sufficient to permit an acceptable game flow while avoiding concerns to the user about the controls. In this scenario, the focus on learning and immersion on the game should reflect on a better performance from the player.

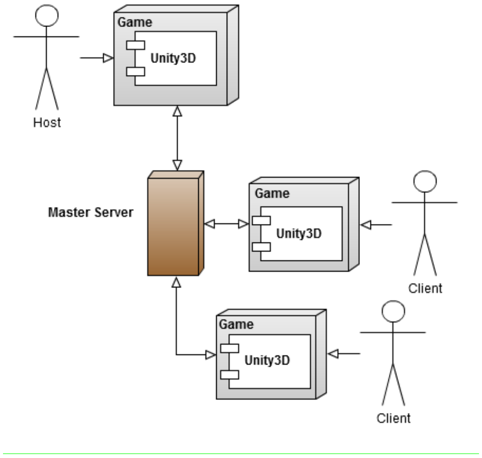


Fig. 1 - Base architecture.

4. Game mechanics

Following the game concept, it is possible to define a rule set that identifies the core game mechanics. Since the game is multiplayer, a Host (Game Master) is required as a point of synchronization for all players. Instantiated cameras and fire extinguishers by God players are saved on XML files and can be loaded at the start of the game. These are called 'Plans'. Only the Host is required to have these plans. The NPC actions can be customized to be able to adapt to specific locations, situations and problems. They can even be customized to the point that they are beyond security issues and adapted to other unrelated problems. These are all saved in XML files and all the players load them from the Host player, similar to the Plan files, increasing this feature's flexibility and ease of use. Besides helping the trainee, these functionalities also provide practical knowledge to supervisors.

4.1. Player types

There are 3 kinds of player that have different roles in the game.

The Security player has the active role of a security professional and is conducting a training plan. This role includes the management of the scene's environment and is the only player who can have a score and feedback at the end of the game.

The Civilian has the purpose to be 'camouflaged' between NPCs and make Security players have more attention to their surroundings. The Civilian player can do all the actions an NPC can do, and even some extra actions if they are defined.

The God player has an important role in the planning and management of the scenario's features. The God player has the ability to create plans, or store in an XML file the positions and attributes of cameras and fire extinguishers that he created on the scene. The God player is the only player that can handle the elements of the scene, being able to instantiate NPC, fires and floods. He may also assume the role of a Security or a Civilian.

4.2. Feedback and score

Points are awarded according to actions and errors performed by the Security players. There are two types of scores: Individual and Team scores.

For the individual score, there are three variables to consider: Difficulty (number of steps to solve the problem), Response Time (time passed since the event was instantiated until it was solved) and Mistakes (mistakes committed while solving the problem). The base score is 1000 points. On each action committed, a score is calculated and added to the base score. The action's score is not always positive so that the final score can be lower than it's initial value. The score is calculated by the following equations:

$$\begin{aligned} &PlayerScore = PlayerScore + EventScore \\ &EventScore = Difficulty * 100 - Mistakes * 50 - ResponseTime \\ &Mistakes = \frac{OverSteps}{2} + MissSteps \end{aligned} \qquad (1)$$

The team score is the sum of the individual scores of the players. Feedback is shown together with the score and the events are presented in chronological order of their resolution, followed by its name. There are three colors associated with the steps taken by the player: Green(Correct step), Yellow(Overstep), Red(Missing step).

4.3. Training Plans and Actions

The training plans are XML files that store information about the positions of the cameras, locations of fire extinguishers, camera attributes, number of NPC in the scene and time to spawn them. These files are created and stored in the system after a God player leaves a match.

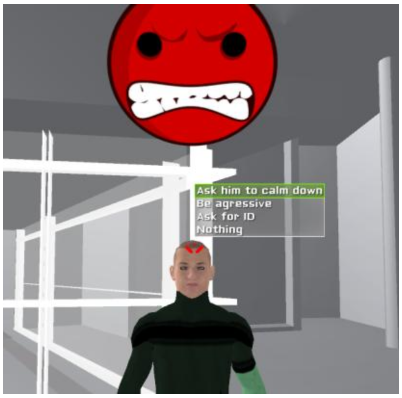


Fig. 2 - Example of an 'Angry' action.

In similarity to the plans, actions are also XML files that are only needed on the Host machine. The actions are customizable and are composed of a set of attributes, including: Name, probability of occurrence (for NPC), name of the billboard that appears above the Civilian that instantiated the action, a set of basic actions (e.g. turn

X degrees) and a set of actions that the security may have and the solution for the event. E.g. Supposing we want to create the 'Angry' with an associated 3\% probability of occurrence the billboard could be a big red angry face. The set of actions that the security player could use can be 'Ask him to calm down!' or 'Be aggressive.' or even 'Ask for ID.'. The solution is always an ordered subset of possible actions. In this case it would be 'Ask him to calm down!' and then 'Ask for ID.'. Fig. 2 shows an example of how this action could be showing on screen.

5. Game development

The game engine used to develop the prototype was Unity3D. This was the best choice for this game due to the possibility of a 3D realistic environment and the engine's low learning curve. The following aspects were analyzed: Architecture, Use Cases, Networking and XML Models.

Events can be actions of a civilian or instantiations of hazards such as fires and floods. Therefore, these are the components that change the score of the players.

In each game, a plan must be chosen or created. These plans are XML files responsible for camera positioning, fire extinguisher positioning and NPC management. The plans are identified by the name of the XML file.

There is a small hierarchy for the players. It is considered that there is a generic 'Player' with some basic functions. This player can be Security or Civilian, and God is both of these.

The player can interact with civilians, close the score screen and even access the pause menu Security players involve specific features: using the camera monitor, change the camera he is viewing on the monitor or change the camera zoom. In the case of fire, the player can pick up an extinguisher to put out the fire (in the case of the use case) and return the extinguisher to its base. The God player covers a wide range of features. Through the HUD (Head-Up Display) options, the player can switch between God and Player Mode. In Player Mode the player is treated as a normal player, while in the God Mode the player is invisible and has no collisions with walls or is affected by gravity. In Player Mode it's still possible to switch between Security and Civilian. In God Mode the player can switch between the camera in first person view or isometric. Assuming the player is in God Mode, you can instantiate the scenario elements that will later be stored in the plans. As for the cameras, the player can instantiate them, delete them and change their properties in real time. For management purposes, the God player also has access the field of view of the cameras. For NPC management, the God player can change the time of instantiation between NPCs and the maximum number of NPCs in the scene. He can also instantiate an NPC in its current position. Like cameras, extinguishers can also be instantiated. These are needed if the God player decides to instantiate a fire.

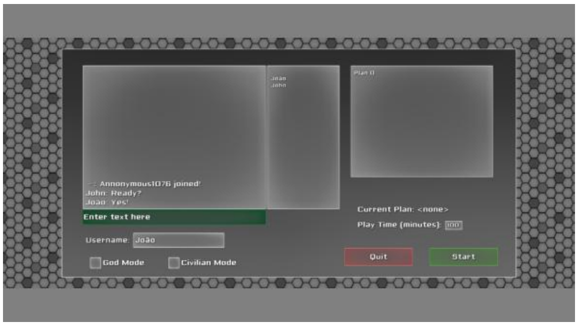
As for network architecture, it is a simple Client-Server structure taking usage of the communication asset between players is the Unity Master Server (UMS). The identification reference sent to the UMS is two strings: the name of the host's game and a comment indicator that tells if the match has started or not.

6. Results and Evaluation

Within the specifications defined in Chapter 3, it is possible to say that most features have been implemented. The major limitation of the prototype is the use of several civilians, since the model used for their representation is quite heavy, even for computers with high performance.

The prototype contains one level: the Faculty of Engineering of the University of Porto (FEUP). The model used is very realistic and represents the real location on a good satisfying level.

A game lobby was also implemented integrating a chat system and gameplay options such as player type and game time. This serves as a meeting point between all players. The result can be seen in Fig. 3.



-: Annonymous1076 joined!
John: Ready?
Joao: Yes!

Joao
John

Username: João

Current Plan: <none>
Play Time (minutes): 100

Fig. 3 - Game lobby where players can meet.

Since the same model was used for the civilians, it was necessary to add something that would differentiate them. A solution for this was the creation of a script that was activated before the NPC was instantiated. This script changed the model's texture regarding his upper body, making it look like they have different shirts. Also, the model's size, walking speed and skin color are randomized.

Due to the multiplayer component of the game, it was necessary to test the game with simultaneous users. Although it is possible to run multiple instances on the same computer, this scenario is unrealistic and does not provide the features to simulate the presence of several players.

As a preliminary evaluation, three students were asked to test the game themselves. While two of the students played on the same game, the other one played alone, alternating between being God player or Security player. The students were asked to fill an inquiry after the tests. The inquiry's questions can be seen on Table 1.

Evaluation results show that overall the students were satisfied with the game, but the biggest flaw is located on the game's feedback functionality as shown in Fig. 4.

Table 1. Inquiry questions. Possible answers ranged from 0 to 10.

Index	Question
1	Is it easy to use the User Interface and to control the character?
2	Are the sounds and models contextualized?
3	Is there a wide range of objects to interact with?
4	Is the feedback regarding executed actions enough?
5	Can the game help the training of security employees?
6	Is God Mode easy to use?

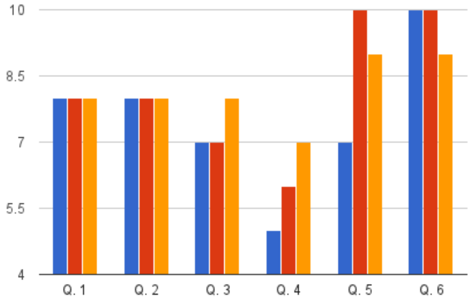


Fig. 4 - Inquiry results. Each colour represents a different student.

As for suggestions, most of the users suggested some kind of different color for the menu that says if the God player is accessing or stopping the player. There was a functionality suggestion, the 'Quick Play' that would let a Host store a previous game configuration (time to play, plan chosen, etc.) and load it instantly.

7. Conclusions and future work

Serious games can be developed for the areas of security and safety to provide more effective training plans. The proposed solution can provide a realistic training environment that is flexible enough to be adaptable to all security companies and locations. The game mechanics is flexible enough to potentiate not only problem-solving skills, but also interpersonal and group effectiveness skills. Teamwork can always lead to new solutions, and therefore, new challenges can arise from supervisors.

Future developments will encompass a thorough evaluation with the contribution of at least one private security company to give constant feedback about the prototypes and methods used to solve problems.

There are also a few aspects to be improved. A visual action creator would be a major feature to help supervisors by providing a real-time representation of the action being created or edited. More content should also be developed with improved design and more levels should be defined.

Acknowledgements

This work is funded (or part-funded) by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project «Eras - Expeditious Reconstruction of Virtual Cultural Heritage Sites - PTDC/EIA-EIA/114868/2009».

References

[1] VSTEP. RescueSim. Available at http://www.rescuesim.com/, last accessed 26th September 2012.

[2] NATO. Boarders Ahoy!. Available at http://serious.gameclassification.com/EN/games/18099-Boarders-Ahoy/index.html, last accessed 26th September 2012.

[3] MAVI interactive. Agent Surefire. Available at http://www.maviinteractive.com/agentsurefire/, last accessed 14th May 2012.

[4] Cryo interactive. Versailles 1685. Available at http://serious.gameclassification.com/EN/games/14672-Versailles-1685/index.html, last accessed 26th September 2012.

[5] Swiss Confederation. International Code of Conduct. Available at http://www.icoc-psp.org, last accessed 26th September 2012.

[6] Confederation of European Security Services. Code of conduct and ethics for the private security sector. Available at http://www.eesc.europa.eu/self-and-coregulation/documents/codes/private/012-private-act-en.pdf, last accessed 2nd June 2012.

[7] E. A. Boyle and T. M. Connolly, T. Hainey and J. M. Boyle. Engagement in digital entertainment games: A systematic review, Computers in Human Behavior, Volume 28, Issue 3, pp. 771-780, 2012. Available at http://www.sciencedirect.com/science/article/pii/S0747563211002640