fAARS: A Platform for Location-Aware Trans-reality Games

Lucio Gutierrez, Eleni Stroulia, and Ioanis Nikolaidis

Computing Science Department, University of Alberta, Edmonton, AB T6G 2E8, Canada
{lucio,stroulia,nikolaidis}@ualberta.ca

Abstract. Users today can easily and intuitively record their real-world experiences through mobile devices, and commodity virtual worlds enable users from around the world to socialize in the context of realistic environments where they simulate real-world activities. This synergy of technological advances makes the design and implementation of *trans-reality games,* blending the boundaries of the real and virtual worlds, a compelling software-engineering problem. In this paper, we describe fAARS, a platform for developing and deploying trans-reality games that cut across the real and parallel virtual worlds, offering users a range of game-play modalities. We place fAARS in the context of recent related work, and we demonstrate its capabilities by discussing two different games developed on it, one with three different variants.

Keywords: Game platform, trans-reality games, virtual worlds, mobile games.

1 Introduction

Location-based games are a type of pervasive games that use the physical space of our entire world as a game board [8]. Their implementation relies on special-purpose middleware that integrates the players' smart-phones so that they can share information about their location, surroundings and actions. The players' game experience can be further enhanced with parallel virtual worlds, so that they can experience the game in the real world and in alternate realities, possibly at the same time. This technological convergence, on one hand, and the pervasiveness of the "gamification" concept across many aspects of our activities today, on the other, has brought to the forefront a new breed of pervasive games called "trans-reality games". This term refers to a relatively new, yet increasingly interesting, class of games, played using mobile devices indoor, outdoor and in virtual worlds. Trans-reality games were first conceptualized by Lindley in 2004. This type of games can be deployed and played in "natural" neighborhoods, such as a campus, a city, or a broader geographical area. Game players interact with the games (a) in the real world through location-specific clues communicated to them through smart-phones that "augment reality" and are aware of each other's locations using GPS or QR codes, and (b) in (possibly multiple) parallel virtual world(s), which represent alternate reality(ies) that reflect the real world in some dimensions and extend it in others.

The fAARS (for Augmented Alternate Reality Services) platform supports the implementation and deployment of trans-reality games in the virtual treasure-hunt and

M. Herrlich, R. Malaka, and M. Masuch (Eds.): ICEC 2012, LNCS 7522, pp. 185–192, 2012.

role-playing style. It consists of a set of components, flexibly integrated with its event-driven game engine, through RESTful APIs. So far, we have developed two different trans-reality games using the fAARS platform. The first game was the "Human Geometric Orientation" game. Through this game, players in a virtual world participate in an experiment, designed to study human orientation mechanisms. The second game, "Outbreak: Safety First", is a "serious game" designed to educate health-science students in precautionary procedures for avoiding nosocomial infections, and can be played in three different game-play modalities.

In the rest of this paper we discuss our work and experience to date. In section 2, we review the background for our work, by reviewing recent trans-reality games and existing platforms for location-aware games. In section 3, the design and implementation of fAARS and its constituent components are fully described. In section 4, we report on our evaluation of the platform through the development of two separate games. Finally, in section 5, we conclude with the contribution of our work.

2 Trans-reality Games and Pervasive-Game Platforms

The "pervasive games" heading includes a broad range of games that integrate ubiquitous technology with physical spaces to provide a gaming experience. Montola [9] introduced a conceptual framework for analyzing the spatial, social, and temporal features of pervasive games, leading to a classification of these games into six types. *Mixed-reality games* are played in physical spaces where virtual and physical components co-exist. *Trans-reality games* are played in physical and virtual spaces at the same time, with the actions of the player avatars in the virtual world affecting the game play in the real world and vice versa. *Adaptronic games* are played in a virtual world but react to real-world events. *Crossmedia games* involve different types of pervasive technologies. *Alternate-reality games* create the illusion that game-play events should be considered "real", thus creating an alternate reality where players act according to hidden clues. Finally, in R*eality games*, everything that happens in the game is real and can have consequences on the real life of players.

Trans-reality games are a particularly interesting and challenging class of pervasive games. "ARQuake" [11] was one of the first outdoor/indoor augmented-reality location-aware games based on the desktop game Quake. Two years later, "Human Pac-man" [4] was the first outdoor augmented-reality game to blend the real and virtual dimensions. Both games required players to wear their particular versions of cumbersome equipment. In 2005, "Magic Land" [12] blended 3D avatars of real players with 3D computer-generated animations, fusing the real and the virtual worlds in a novel way. Between 2004 and 2006, trans-reality games, such as "Can you see me now?" [1], "Uncle Roy All Around You" [2], and "I Like Frank" [3], demonstrated the use of mobile devices in real-world playgrounds blurred with virtual worlds, with players in both worlds advancing the game, although the actions allowed to players in the virtual world were very limited. These games were developed on special-purpose hardware and software technologies, with no reuse potential. Furthermore, in the above trans-reality games, virtual worlds have been used primarily as an information resource, and there is still substantial work to be done towards the comprehensive integration of virtual- and real-world game-play.

Table 1. Pervasive-Games Platforms

PLATFORM	Indoors	Outdoors	Individual	Teams	Map	Augmented Browser	Raw Game Data	Treasure Hunt	RPG	Capture-the-flag	Peer-to-peer	Client-server	Game Authoring
fAR-Play [6]	√	√	√			√	√	√				√	√
SCVNGR - www.scvngr.com		√	√		√		√	√				√	√
MUPE [13]		√	√				√	√	√		√	√	
Mobile Chase [5]		√	√				√	√				√	
FRAP [14]		√		√	√		√	√		√		√	
Wherigo - www.wherigo.com		√	√		√		√	√				√	√
PCAFPEA [10]		√	√		√	√	√	√			√	√	

As fAARS is on the overlap between trans-reality and location-aware mobile games, we review above the features of some important platforms for location-aware games. The comparison of the features of these platforms is summarized in **Table 1.** As it can be seen in the table, a few of the platforms support indoors and outdoors game-play but none support virtual worlds; thus, they are not able to support trans-reality games. Most are tailored to individual game-play, except from FRAP that also supports teams. Most augment the players' game-play experience with game-related content; some include an augmented-reality browser that superimposes this content on the image that players see through their cameras, or on a map where they can see their (and other players') location(s). All support treasure-hunt games; Mupe also supports role-playing games and FRAP supports capture-the-flag games. All support communication between players through the game engine; Mupe and PCFPA also support peer-to-peer interactions of players in close proximity to each other. Finally, most of the platforms offer some game-authoring tools.

3 The fAARS Software Architecture

fAARS was conceived as a new-generation pervasive-game platform, extending the space of the game play from the real world (indoors and outdoors) to parallel virtual world(s) in order to support location-aware trans-reality games. In these games (a) the real-world space and locations (and any information associated with them) are mapped to the virtual world; (b) the movement and actions of the real-world players is reflected through their avatars in the virtual world; (c) the movement and actions of the virtual-world players is communicated to real-world players through their smart phones; and (d) in addition to real-world and virtual-world players, non-playing characters, controlled by a simulation engine, can also change the game state. fAARS supports the deployment of trans-reality games in the virtual treasure-hunt and role-playing game styles. fAARS integrates OpenSim (http://opensimulator.org/) as the virtual-world game-play environment and considers two types of mappings between the real- and virtual-world spaces. In the "parallel worlds" mapping, the virtual-world setting mimics the real-world setting and the coordinates of every location in the real-world playground are transformed to a pair of virtual-world coordinates. In the "points-of-interest" mapping, the virtual world may represent a conceptually different metaphor from the real world, and only specific points of interest may be mapped across the two. In this case, the players' locations can only be reflected across the two worlds when the players are close to any of the pre-defined points of interest.

fAARS supports three different game-play modalities [7]. In the *Dormant* modality, players play the game in the real world using smart-phones. In the *Dreaming* modality, players play the game in the virtual world. Finally, in the *Astral-projection* modality, players choose to play in the virtual world or in the real world using smart-phones; in the latter case, the real-world players' avatars interact with other players and objects in the virtual world at the same time.

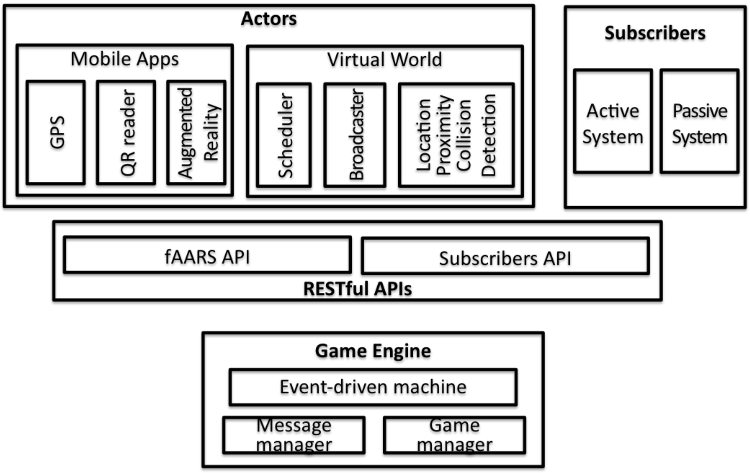

Fig. 1. The fAARS Platform Software Architecture

As shown in Fig. 1. , the fAARS platform consists of two main components: (a) the *Actors* (and their devices) corresponding to the real-world and virtual-world players; and (b) the *Game Engine*, which is responsible for recognizing the *Actors'* actions and inferring the next game state based on the game rules. The *Actors* access the game engine through the *fAARS API*. Furthermore, external components (i.e., Subscribers) can be integrated with the *Game Engine* through the *Subscribers API*. The fAARS platform is implemented according to the Event-Driven SOA (Service Oriented Architecture) style.

The Actors: In fAARS, every player in the real and virtual worlds is represented as an *Actor*. In the real world, *Actors* interact with the game using a mobile application running on a smart-phone. The mobile application senses the *Actors'* location and actions, using the smart-phone GPS and a built-in QR-code reader. In the virtual world, players have the same cross-platform application running on a HUD (Heads-Up Display). The HUD senses the virtual-world through collision, proximity, and location detectors. The virtual world also provides a *Scheduler* and a *Broadcaster*.

The former supports the communication between the virtual world and the game engine, through a XML-RPC port, and maintains a set of global parameters related to the virtual-world game. The *Broadcaster* maintains a list of the active virtual-world objects and forwards them relevant communications from the *Game Engine*.

The Game Engine is responsible for maintaining and advancing the state of the *game*; it receives *Events* from the *Actors*, processes the game rules and accordingly updates the state of the game, and notifies the players and registered *Subscribers*. The *Game Engine* consists of (a) the *Event-driven Machine*, (b) the *Message Manager*, and (c) the *Game Manager*. The *Message Manager* receives *Actor*-generated events and forwards them to the *Event-driven Machine,* which communicates with the *Game Manager* to query the fAARS database for the relevant game rules[1]. Based on the rules, the *Event-driven Machine* updates the state of the virtual world and the state of the *Actors* and notifies the *Subscribers* about *Events* of interest. The *Message Manager* notifies the *Actors* as necessary and updates the cross-platform mobile applications and the virtual-world HUDs, without requiring players to explicitly request updates from the *Game Engine*.

The Subscribers RESTful Interface: fAARS provides a REST service that allows external systems to observe a game and to provide further downstream functionality to fAARS games, as if they were part of the platform. As shown in Figure 1, *Subscribers* can be either Active or Passive. Active subscribers can push events to the *Game Engine* (through the *pushEvents* API) where passive systems are only notified about *Events* (through the *registerObserver* API).

The fAARS RESTful API is composed of thirty APIs, which provide access to functionality of different internal pieces of fAARS, specifically to the internal pieces of the *Game Engine* as depicted in Figure 1. The APIs are divided in four categories: a) event-driven machine (push events: 2 APIs), b) game manager (query the game state: 16 APIs), c) message manager (send and receive messages: 2 APIs), and d) virtual world (update the virtual world: 10 APIs).

4 Two Games on the fAARS Platform

We have validated the usefulness and generality of fAARS by developing two games with it.

The “Human Geometric Orientation” Game (HGO): Systematically examining whether and how each geometrical property impacts animals and humans' orientation implies the need to “build” environments in a controlled manner with their properties systematically varied. This is practically impossible and this is why our psychology colleagues decided to conduct their study by developing a fAARS game. In this game, players were trained to locate two geometrically equivalent corners in a number of parallelogram-shaped rooms in a first-person navigable virtual world. With the HGO

[1] Note that, as of today, fAARS does not offer an authoring environment; therefore, the game rules have to be stored directly in the database by someone familiar with the fAARS database design.

game, the researchers could systematically specify the types of room shapes they wanted, and we were able to systematically construct them in the virtual world using the OpenSim building tools. Of course, it was impossible to construct parallel real-world settings; therefore, the HGO game was played only in the virtual world, in the so-called *Dreaming modality*, with players logging in the virtual world and interacting with the room corners to generate virtual-world *Events*.

Referring to Fig. 1. , the fAARS components used for the HGO game are the virtual world, the game engine, and an external system as an active subscriber.

- A total of 99 *Actor* instances (one for each subject) with unique IDs were created to allow participants to interact with the corners in each of the rooms in the virtual world. The room corners were also represented as *Actor* instances, in order to rely on the virtual-world collision detection for generating corner-choice events
- The HGO *game engine* has a single rule that upon recognizing a corner-choice event, it increases the score of the *Actor,* if the corner choice was correct.
- The *Virtual World* was the setting of the game with rooms constructed manually in a systematic manner and the corners associated with invisible *Actors*.
- There is a single *active event subscriber* that keeps track of the number of tries of every *Actor* for each of the rooms in the virtual world, and updates the location of the *Actor* instances based on the number of tries.

The "Outbreak: Safety First" Game (Outbreak): The second game developed on fAARS, "Outbreak: Safety First", was designed as a learning activity for medical students in an epidemiology course. The game narrative places the players in a hospital where they have to visit a number of patients. As they move from one patient to another they have to avoid infectious viruses and to answer questions. At the patients' rooms, they have to follow appropriate precautionary procedures to protect themselves from infections, depending on the patient symptoms.

Taking full advantage of the fAARS functionalities, this game was deployed in three different variants. The first variant was played in the real world only, in a teaching-hospital ward. Players have to scan QR codes to see the symptoms of the simulated patients they visit, to select the appropriate gear to don, to answer questions, and to get treated if they were infected. As the game evolves, players "get infected" through actions dictated by the game (scanning QR codes attached on infected landmarks) or through the exposure of their avatars to virtual viruses in the simulation running in the virtual world. When this happens, they are informed through a message to their mobile device, at which point, they have to choose appropriate measures to protect themselves and the patients. The viruses, their infection cycle, their contagion to patients and players are controlled through a simple simulation in the virtual world, setup in a parallel mapping to the real world. This type of game play represents the so-called *Astral-projection modality,* where virtual- and real-world players can interact with each other in any world, and activities are reflected in both worlds.

The other two game *Dreaming modality* variants were deployed in a virtual world, as desktop video games. The first one uses the virtual world as a clickable 2D map (virtual 2D version) of a Pacman board, on which players and viruses interact with each other. The third game variant takes place in a 3D virtual world, where the virtual space configuration is similar to the ward of the mobile version, and players have a first-person view in a navigable virtual world.

The first game variant uses all the fAARS components, including two external systems as active subscribers. The other two game variants use the virtual-world actors, the game engine, and the same two external systems as active subscribers (similar to the HGO game). Let us now review the development process for the Outbreak game.

- We created *Actor* instances for each of our student participants; some were real-world Actors (for the first game variant) and some were virtual-world Actors (for the other two variants). In addition, a number of *Actors* acting as *NPC*s (Non-player Characters) were deployed to represent the virtual-world viruses.
- The Outbreak *game engine* has many more rules than the HGO game that control the interactions of *Actors* with the QR codes representing questions, the QR codes outside the patients' rooms, and the collision with viruses. The *Event* types used to define the *ECA* rules of the game ware the *decodeQR* and *onCollision Events*.
- The *virtual world* was an essential component for all three versions of this game. In the *Astral Projection* modality, the virtual world is used as an extension of the real world, where all *Actor* instances can be infected by *NPCs* representing viruses flying around in the game. In the *Dreaming* modality, it is in the virtual world where the interactions among all *Actors* and *NPCs* happen during game-play.
- There were two *event subscribers* in this game that provided further downstream functionalities for all three versions of the game. The first was a repository of questions, which allowed players to increase their score if they correctly answered a question. The second was a mobility controller that moves the virus *NPCs* in the virtual world in a pseudo-random fashion.

5 Conclusions

The development of fAARS was motivated by our realization that, with the increased maturity and adoption of mobile and virtual-world platforms and momentum of the *"gamification"* paradigm in education there is a great opportunity for trans-reality games. The fAARS platform supports a specific style of trans-reality games, grounded on a complex spatial metaphor and following the virtual treasure-hunt and RPG styles. The fAARS platform extends the state-of-the-art in pervasive location-aware gaming platforms in the following ways.

fAARS is the only platform that enables the deployment of location-aware games in a virtual world, as an extension of the physical indoors/outdoors playground. The virtual-world playground may "parallel" the real world (with all real-world and virtual-world coordinates mapped through a bi-directional function) or it may communicate a completely different metaphor to the game story (mapping only interesting locations across the worlds). fAARS supports peer-to-peer interactions among players in the virtual world. Mimicking the functionality of NFC, a special-purpose software device in the virtual world recognizes player proximity. This event can be used by the game story to notify the players and give them opportunities to advance the game play. Finally, fAARS enables a variety of game play combinations in the real and the virtual world. The two example games discussed in Section 4 demonstrate how fAARS can be used to develop real-world only games (like the platforms discussed in Section 2.), or virtual-world only games, or true trans-reality games.

References

1. Benford, S., Crabtree, A., Flintham, M., Drozd, A., Anastasi, R., Paxton, M., Tandavanitj, N., Adams, M., Row-Farr, J.: Can you see me now? ACM Trans. Comput.-Hum. Interact. 13, 100–133 (2006)
2. Benford, S., Flintham, M., Drozd, A., Anastasi, R., Rowland, D., Tandavanitj, N., Adams, M., Row-Farr, J., Oldroyd, A., Sutton, J.: Uncle Roy All Around You: Implicating the City in a Location-Based Performance. In: Proc. Advances in Computer Entertainment (ACE 2004), ACM Press (2004)
3. Blast Theory (n.d.). I like Frank, http://www.blasttheory.co.uk/bt/work_ilikefrank.html (retrieved April 1, 2012)
4. Cheok, A.D., Goh, K.H., Liu, W., Farbiz, F., Fong, S.W., Teo, S.L., Li, Y., Yang, X.: Human pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. Personal Ubiquitous Comput. 8 8, 71–81 (2004)
5. Fetter, M., Etz, M., Blechschmied, H.: Mobile chase–towards a framework for location-based gaming. In: GRAPP (AS/IE), INSTICC–Institute for Systems and Technologies of Information, Control and Communication, pp. 98–105 (2007)
6. Gutierrez, L., Nikolaidis, I., Stroulia, E., Gouglas, S., Rockwell, G., Boechler, P., Carbonaro, M., King, S.: far-play: A framework to develop augmented/alternate reality games. In: IEEE PerCom Workshops, pp. 531–536 (2011)
7. Lindley, C.A.: Game space design foundations for trans-reality games. In: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, ACE 2005, pp. 397–404. ACM, New York (2005)
8. Magerkurth, C., Cheok, A.D., Mandryk, R.L., Nilsen, T.: Pervasive games: Bringing computer entertainment back to the real world. ACM Computers in Entertainment 3(3), Article 4A (2005)
9. Montola, M.: Exploring the Edge of the Magic Circle. Defining Pervasive Games. In: Proc. of Digital Experience: Design, Aestethics, Practice Conference, Copenhagen (2005)
10. Papakonstantinou, S., Brujic-Okretic, V.: Prototyping a context-aware framework for pervasive entertainment applications. In: Proceedings of the 2009 Conference in Games and Virtual Worlds for Serious Applications, VS-GAMES 2009, pp. 84–91. IEEE Computer Society, Washington, DC (2009)
11. Piekarski, W., Thomas, B.: Arquake: the outdoor augmented reality gaming system. Commun. ACM 45, 36–38 (2002)
12. Qui, T.C.T., Nguyen, T.H.D., Mallawaarachchi, A., Xu, K., Liu, W., Lee, S.P., Zhou, Z.Y., Teo, S.L., Teo, H.S., Thang, L.N., Li, Y., Cheok, A.D., Kato, H.: Magic land: live 3d human capture mixed reality interactive system. In: CHI 2005 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2005, pp. 1142–1143. ACM, New York (2005)
13. Suomela, R., Räsänen, E., Koivisto, A., Mattila, J.: Open-Source Game Development with the Multi-user Publishing Environment (MUPE) Application Platform. In: Rauterberg, M. (ed.) ICEC 2004. LNCS, vol. 3166, pp. 308–320. Springer, Heidelberg (2004)
14. Tutzschke, J.-P., Zukunft, O.: Frap: a framework for pervasive games. In: Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2009, pp. 133–142. ACM, New York (2009)