Rapid Prototyping for Mobile Serious Games

José Rouillard[1], Audrey Serna[2], Bertrand David[3], and René Chalon[3]

[1] LIFL Laboratory, University of Lille 1,
F-59655 Villeneuve d'Ascq Cedex, France
jose.rouillard@univ-lille1.fr
[2] Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205,
19 avenue Jean Capelle, F-69621 Villeurbanne Cedex, France
audrey.serna@insa-lyon.fr
[3] Université de Lyon, CNRS
Ecole Centrale de Lyon, LIRIS, UMR5205,
36 av. Guy de Collongue, F-69134 Ecully Cedex, France
{bertrand.david,rene.chalon}@ec-lyon.fr

Abstract. Mobile Serious Games are new kind of Serious Games which are running on mobile devices, mainly on Smartphones. With continuously increased power and User Interface facilities, they constitute an alternative to the usual entertainment applications proposed on Smartphones. To design and implement such applications, a methodological assistance and development support are required. In this paper, we present our contribution to rapid prototyping for Mobile Serious Games in which we propose to augment *App Inventor for Android* framework with a methodological assistance. This proposition is based on a study in which we asked to 116 students to use this framework for the development of mobile applications. The results are presented (thematic domain, targeted users, components used...) and we discuss the relevance of using such a tool to achieve rapid prototyping for mobile Serious Game.

Keywords: Human-computer interaction, Serious Games, Mobile learning, Prototyping, App Inventor.

1 Introduction

To use mobile devices, as smartphones, for other purpose than entertainment seems an interesting orientation in order to exploit the small size of these devices and their contextualization facilities, and user's availability anywhere (in transportation, in waiting moments, ..) to devote this time to a more helpful activity, as learning. Mobile Learning or M-Learning refer to this kind of learning, which can be either context independent, using user's availability to learn in any location or context dependent, taking into account user's location (geographical, logical, …).

Serious Games (SG) are pedagogical games that educate, train and inform ([2], [3], [6], [7], [9]). SG were initially introduced to support sophisticated learning in

P. Zaphiris and A. Ioannou (Eds.): LCT 2014, Part II, LNCS 8524, pp. 194–205, 2014.

contextual situations in which learners play their roles in an authentic scenario (or working situation) expressed by a simulator. Company management investments or gaining new market places are archetype scenarios used for SG. For this kind of SG a well-organized design process allowing team working is needed [4], as well as a development infrastructure, an IDE, allowing developing a simulator and user interfaces for all users [1]. These heavy applications require usually several years of developments [8].

In the following section, we present related work and state of the art around the notion of SG. We present next how to expand an existing framework, App Inventor for Android, for rapid prototyping for mobile SG.

2 Previous Work on Designing Serious Games

2.1 Situated and Mobile Serious Games

Previous works conducted at LIRIS laboratory are aimed to design learning environments, in particular SG, which are supported by the use of technologies and that relied on situated learning theories.

From a general point of view, some guidelines are aimed to design situated learning environments [10], such as providing authentic context and activities and supporting learners' collaboration, reflection and articulation for the construction of knowledge and abstractions. In addition, the learning environment should allow learners to observe different aspects of a situation by adopting different roles or perspectives.

SGs meet intrinsically a part of situated learning environments requirements since they usually integrate role-playing aspects and provide coaching and feedback elements embedded into the activities. In addition, the advances of technologies in the field of HCI can foster the authenticity of activities and offer tools to help learners in reflecting on their actions, reasoning and building their knowledge together. Mobile technologies offer interesting perspectives thanks to their characteristics such as connectivity, mobility and context sensitivity. Mobile SG (M-SG) can be able to extract, interpret and use contextual information in order to adapt its content to the authentic context. Other features, such as augmented reality, can be easily deployed on recent mobile devices, and can be used to favor learning professional gestures in a real-life situation (learning by doing paradigm).

In previous work [1], we argued that these technologies, in particular mobile devices, should be integrated into more global learning systems forming a set of heterogeneous platforms. This point of view addresses several issues at different granularity levels. For example, at the highest abstraction level, the learning game should be able to adapt its content to each learner experiences. At a lower level, the combination of platforms should be managed dynamically: the system should be able to distribute the users interfaces (redistribution mechanisms [11]) in order to increase collaborative and authentic aspects.

To explore these concepts, we proposed different scenarios integrating mobile or collaborative aspects in global learning environments (SEGAREM project described in next section). We also explored lighter applications using scenarios [1] and paper-based prototypes at the earlier stages of a user-centered design approach [12] to measure the collaborative potential of a learning scenario using tangible interactions on multitouch tabletops. This evaluation gave us some hints on the design quality but to go further in the evaluation, we need to implement the game and test some synchronization aspects.

2.2 SEGAREM Project and the Le(a)rnIT Prototype

The implementations of SGs are highly cost and time consuming, and rapid prototyping tools are necessary to develop and test different solutions to foster collaborative and contextualized activities. In SEGAREM project (which lasted 3 years – 2010-2012) we developed a SG prototype called Lear(n)IT aimed at teaching the Lean Manufacturing methodologies [13]. Players' goal is to manufacture as many products as possible in a limited production time. Each learner plays an operator role in the industrial production line to understand the complexity of its dynamicity and how to improve it. Raw materials and processed materials are moved between the player's tables by a warehouseman handling a cart. After each simulated working sequence, the teacher and learners debrief their working experience in order to find improvements to apply to the production line (as presented in the lectures).

From the hardware perspective (see Figure 1), three tangible interface-supported tabletops (MT1 to MT3), a Samsung Surface 2 tabletop, an Android Tablet (Ta), an Android smartphone, and classical personal computers compose the learning environment. The Tablet and Smartphone are used as mobile interfaces to convey raw materials and processed products, while tabletops (workstations) have a specific position defined by game rules and in-game debriefings. The spatial configuration between tabletops is very important in defining object flows and is crucial to the optimization process.

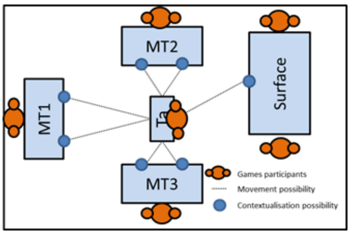


Fig. 1. Physical arrangement of participants and devices in Lear(n)IT

From an interaction perspective, the production line is supported by 4 augmented tabletops and a tablet. Each workstation waits for material to enter its input area in order to transform it. These materials are moved between tables using the tablet carried by the learner playing the warehouseman role. Material processing tools are represented by tangible interfaces directly on each table, where materials are also digitally represented.

In SEGAREM we designed several toolkits to develop the prototype, which can be reused for helping in developing further projects. Despite these toolkits, developing such a complex serious game which involves several users and devices is a very time consuming task and required high programming skills. Rapid prototyping tools are necessary to develop and test different solutions to foster collaborative and contextualized activities.

3 Mobile Serious Game Design and Implementation Support

The SGs for mobile devices are a transposition of initial SG orientation taking into account new characteristics related to mobility. If SGs and M-SGs are known to help learners develop specific skills, they are not so widespread mainly because of implementation cost. "Their use has proven to be promising in many domains, but is at present restricted by the time consuming and costly nature of the developing process." [8].

Basically, we can consider two classes of SGs: complex SGs and simpler Mobile SGs. Complex SGs require a development by teamwork with specialized actors, complex simulator and have a significant cost. Aldrich considers the estimated cost of this kind of SGs between 10 and 300 thousand dollars [15].

M-SG must be light, at first individual with short sessions, but progressively also multiplayers. In this way, the design and development of M-SG is different from conventional SG design and implementation. Individual design and implementation are prevailing, in order to privilege innovation and creativity. Designers, who are neither design specialists nor experienced implementers, should be assisted in these activities (designing and programming) by providing methodological and implementation supports. Thus, this research focuses on:

- Concerning programming issue, visual programming by assembly various components is an interesting and proven approach: it can be seen as an end-user programming method.
- Concerning methodological issue, the architectural approach allows us to propose a model of thinking based first on an interactive application structure (architecture) organized in 5 categories of components: HCI, contextualization, data management, treatment and communication, then its increasing by introduction of two SG oriented components: SG springs and SG engines.

We are going to present in the following section how App inventor can be used in such context and what are the strengths and weaknesses of this tool for developing SGs.

3.1 Developing Serious Games with App Inventor

From a development point of view, rapid prototyping and end-user programming are interesting approaches to speed up and increase the quality of produced applications.

App Inventor for Android framework is a tool for rapid prototyping based on visual programming. It is aimed to facilitate the design and deployment of Android applications. The main advantage to use App Inventor for Android is that actors are not necessarily supposed to be computer scientists or programming developers. By manipulating (drag and drop on visual blocks) components from palettes and snapping them together, this Google and MIT tool allows to generate usable and powerful applications, that can be used, for instance, in the SG's world.

The App Inventor design editor is composed of 9 palettes of components that are the following (in version 2): User interface, Layout, Media, Drawing and Animation, Sensor, Social, Storage, Connectivity, Lego® Mindstorms®. It allows developers not to spend too much time learning programming skills (Java), but instead, to explore technologies actually related to the mobility of users and ergonomic matters.

We tested this framework with a consequent group of students, who programmed in a short period devoted to class works a lot of interactive applications, games and serious games.

3.2 Experimentation

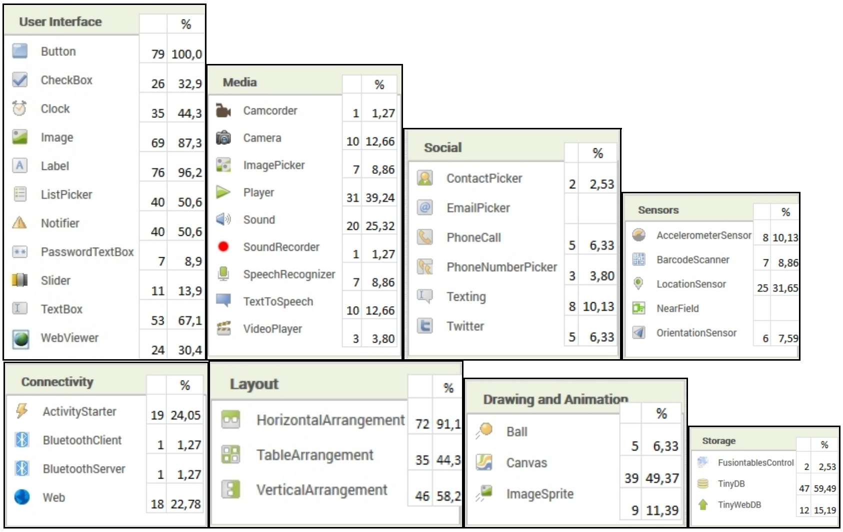
We have conducted a study involving 79 projects, developed by 116 students from 5 different classes. Students were asked to use App Inventor to develop mobile applications, games and serious games on Android smartphones and tablets.

Among those 116 students, 60 (51.72 %) had a personal Android smartphone, and 8 (6.90 %) had already developed a mobile application. The students worked at least during 10 hours (5*2 hours) at university, to produce the final version of their mobile application, with the advices of a teacher. They used ACER Stream, Liquid MT and Z2 Duo Android smartphones to test and package their applications.

The topics of the mobile applications developed by the students were mainly oriented towards tools & utilities, games & entertainment, social, sport, and travel aspects. The categories of targeted users chosen by the students are the following: All public (46.84 %), Adult (44.30 %), Other (24.05 %), Adolescence (15.19%), Childhood and pre-adolescence (1.27%). The category "Other" was used to specify particular users such as disabled people, professionals, etc.

Figure 2 presents the palettes of component used by the students, in order to develop their mobile applications. For each palette, the first column indicates the number of projects (among 79) that used this component, and the second column is the percentage associated. For example, as we can see, 79 Android applications (100% of the projects) used the Button component

The great majority of the developed mobile applications are non-collaborative (single user mode). Only one project used a Bluetooth communication in order to play a collaborative pong game, and 12 projects used a TinyWeb component, allowing data exchanges on the Internet network.



User Interface		%
Button	79	100,0
CheckBox	26	32,9
Clock	35	44,3
Image	69	87,3
Label	76	96,2
ListPicker	40	50,6
Notifier	40	50,6
PasswordTextBox	7	8,9
Slider	11	13,9
TextBox	53	67,1
WebViewer	24	30,4

Media		%
Camcorder	1	1,27
Camera	10	12,66
ImagePicker	7	8,86
Player	31	39,24
Sound	20	25,32
SoundRecorder	1	1,27
SpeechRecognizer	7	8,86
TextToSpeech	10	12,66
VideoPlayer	3	3,80

Social		%
ContactPicker	2	2,53
EmailPicker		
PhoneCall	5	6,33
PhoneNumberPicker	3	3,80
Texting	8	10,13
Twitter	5	6,33

Sensors		%
AccelerometerSensor	8	10,13
BarcodeScanner	7	8,86
LocationSensor	25	31,65
NearField		
OrientationSensor	6	7,59

Connectivity		%
ActivityStarter	19	24,05
BluetoothClient	1	1,27
BluetoothServer	1	1,27
Web	18	22,78

Layout		%
HorizontalArrangement	72	91,1
TableArrangement	35	44,3
VerticalArrangement	46	58,2

Drawing and Animation		%
Ball	5	6,33
Canvas	39	49,37
ImageSprite	9	11,39

Storage		%
FusiontablesControl	2	2,53
TinyDB	47	59,49
TinyWebDB	12	15,19

Fig. 2. Components used by the students in their mobile applications

It's interesting to notice that various sensors were employed by student developers: 25 projects among 79 (31.65 %) used a "LocationSensor" component (GPS), 8 (10.13 %) used an "AccelerometerSensor", and 6 (7.59 %) used an "OrientationSensor" (compass). Figure 3 shows the blocks needed to detect automatically the location of the user, and the resulting interface of this implementation on a real Android device.

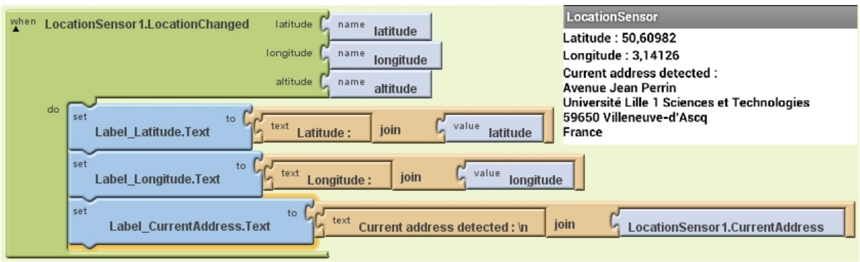


Fig. 3. Detecting latitude, longitude and current address, with the location sensor component of App Inventor

Among the 79 projects, we consider that 14 applications can be seen as prototypes of interesting M-SG, and we are focusing on those projects in the following section.

3.3 Focus on 14 M-SG prototypes

Those M-SG prototypes are applications that are really connected to the user environment and that used various components, such as barcode scanner, accelerometers, compass, shake detection, etc. In such applications, users can play and learn, by interacting with the (serious) games.

Figure 4 (left) presents an application called “I learn the numbers” in which the children have to enter (touch canvas components) a number pronounced by the machine with a Text-To-Speech synthesis. Figure 4 (center) presents a game to learn various things about horses (how to recognize them, how to care and feed them, etc.). Figure 4 (right) is a game developed to learn English: the users have to touch the right number pronounced by the digital professor, etc.

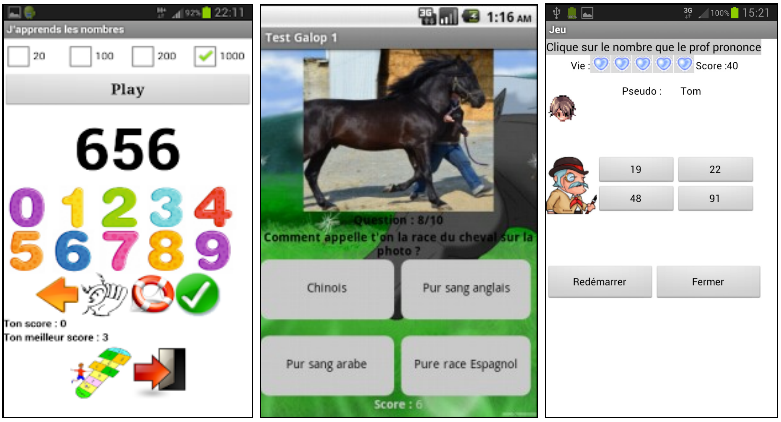


Fig. 4. Examples of Serious Game developed with App Inventor

Figure 5 is an example of application developed by students with App Inventor, for fighting against Alzheimer’s disease. Some words are presented, textually and vocally, to the user (left). Then a parametered countdown is displayed (center). Finally, the user is asked to enter in a textbox the words that s/he can remember, and the smartphone indicates the score of the user (right). Concerning the development of those 14 M-SGs, the used components are distributed in five groups, as the following:

- HCI : Buttons (100 %), Horizontal Arrangement (100%), Labels (86 %), Images (86 %), Vertical Arrangement (71 %), Canvas (64 %), Player (57 %), Table Arrangement (50 %), Textbox (50 %), Notifier (50 %), Password (43 %), Sound (36 %), Checkbox (35 %), TextToSpeech (29 %), WebViewer (29 %), ImageSprite (29 %), Slider (27 %), ListPicker (14 %), Image Picker, VideoPlayer and Ball (7 % each);
- Contextualization: Location (29 %), Accelerometer (21 %), BarcodeScanner (7 %) and Orientation (7 %);
- Treatment: Clock (43 %);
- Data management: TinyDB (50 %), TinyWebDB (14 %) and FusionTableControl (7 %);
- Communication: Web (14 %) and PhoneCall, (7 %).

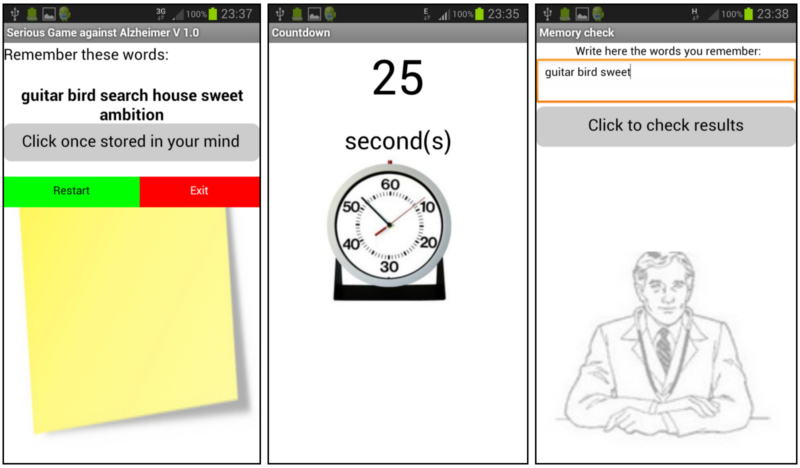


Fig. 5. Example of serious game for fighting against Alzheimer's disease

4 Facilitating the Implementation of Mobile Serious Games

From a methodological point of view, it seems important to indicate that an analysis of the target interactive application can be driven by its architectural structure, which is mainly based on five aspects: HCI, Sensors, Data management, Communication and Treatments (computational behavior of the application). These five architectural aspects are not too far from components categories of App Inventor framework. In this way the mapping between these two views is relatively easy.

In order to increase applicability of App. Inventor framework, three directions can be explored:

- Elaboration of new components, which increase the scope of application behaviors. Main orientation seems be "Treatment components", i.e. reusable computational behaviors to be used in future applications;
- Creation of composite components obtained by interconnection of existing components, and use of them as basic components;
- Merge of existing applications and manipulation of their components in order to create new applications.

4.1 Facilitating the Implementation of Mobile Serious Games with App Inventor

Figure 6 presents the "421 game", developed with App Inventor. Each time the user touches a dice, a random number procedure is invoked and the result is displayed in the calling application.

Fig. 6. Example of "421" game using our "roll the dice" procedure

When the random number is chosen, x milliseconds (see parameter) are awaited before that secondary application returns the response to the primary application, and kills itself. In this example, the procedure was called three times and the resulting number obtained is 316. Figure 7 shows the blocks needed to invoke the "Roll the dice" procedure.

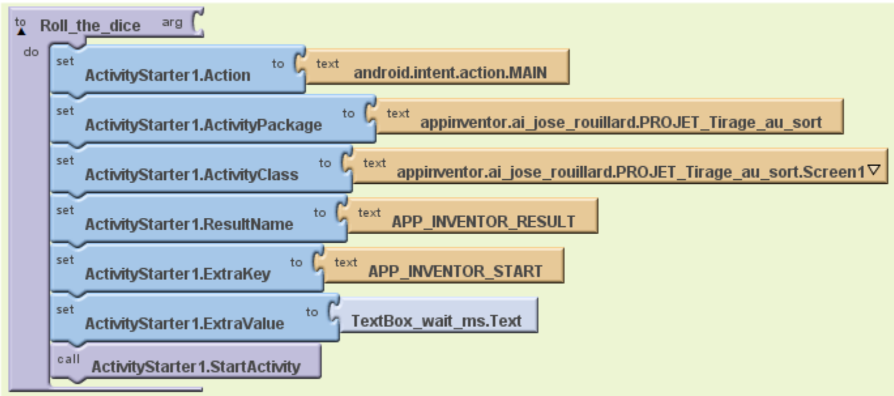


Fig. 7. The "Roll the dice" procedure is calling another activity with parameters, to get a result

With this kind of patterns and pre-programmed procedures, we believe that developers of M-SG would be more efficient with App Inventor. It would reduce their development time and would help to generate some safe and robust M-SG.

4.2 Going Further Towards Mobile Serious Games

App Inventor provides a range of interesting components in these 5 categories: HCI (Input, Output, Layout, animation), contextualization (sensors), data management (local or web DB), communication and collaboration (social aspects), but it offers relatively limited components for treatments and meta-treatments. To go further towards M-SG development, it is interesting to take into account the following three additional aspects: game principles, game engine and more complex treatments.

Regarding game principles, Mariais and al. [14] propose some motivating factors and important characteristics to take into account in SG:

- Being in competition (form of competition: independent actions, sequential, data, time or position actions, calculation type of victory)
- Playing a role (special abilities associated with a role)
- Being subject to chance (the impact of involving an element of chance)
- Managing a high-risk situation (qualification of the crisis situation)
- Acting collectively (choice of cooperation / collaboration methods)
- Receiving recognition (type of information sharing, personalization, feedback/notifications)

Regarding the collaborative aspects of M-SG, it could be interesting to model game playing behaviors in order to provide a game playing management engine. We identified at least 6 game playing behaviors, which can be expressed by corresponding engines:

1. No constrained individual actions: the users (players) are able to work separately without coordination and data evolution: exploration of a compartmented universe;
2. Sequential game between players: each player can action only one in a predefined order;
3. Time organized players' participation (playing schedule);
4. Data constrained players' participation managed by data accessibility and update;
5. Real time synchronization, update of data and players locations;
6. Management of team of players' game participation with cooperation.

These engine behaviors can either be proposed as new components, in relation with previous extension suggestions, or as component patterns, which can be recomposed during the game development.

Finally, for complex treatments, a set of new "Treatment components" can be added to App. Inventor in order to facilitate their integration by composition. We are proposing an open-ended list of fairly generic treatments:

- Score, ranking and awards calculation;
- Interval related random calculation;
- Timer, scheduler and overall time management;
- Choosing characters and avatar routines;
- General and particular calibrations (user profile, sensors' accuracy, etc.).

5 Conclusion

In this paper we have shown the relevance of using App Inventor to achieve rapid prototyping for M-SGs. We have observed that, without coding in Java, 116 students succeeded in developing 79 real prototypes (APK easily generated for Android smartphones and tablets) in a short time (around 10 hours). We focused our study on 14 mobile applications that can be considered as interesting M-SG. Indeed, the user can really learn relevant information by playing and interacting with those prototypes.

In the design and development of M-SGs it is very important to detect the user's context and to offer information and data related to this context (location, orientation, time, device's features…). With such kind of tools, it becomes relatively easy for designers to integrate mobile applications features and sensors, such as GPS, barcode scanner, accelerometer, orientation sensors, video/camera recorders, etc., that certainly improve the usability of the developed M-SGs. The weaknesses of using App Inventor for M-SGs are related to the poor possibilities to reuse existing blocks and/or patterns already developed in other projects.

Our main contribution was to propose a methodological helping for the design and implementation of M-SGs designed with App Inventor. We illustrated this possible reuse of patterns with an example (Roll the Dice) invoking a StartActivity component with parameters.

In the close future, we will provide more M-SGs patterns and/or super-components, in order to facilitate the implementation of applications related to pedagogical and learning concerns. We will also work around the notion of context awareness and pedagogical learning style available within tools like App Inventor, in order to improve the tutoring activity and the collaboration in M-SGs. Our next job will be to measure and evaluate the usability of M-SGs developed with App Inventor enhanced with the propositions made in this work.

References

1. George, S., Serna, A.: Introducing Mobility in Serious Games: Enhancing Situated and Collaborative Learning. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part IV, HCII 2011. LNCS, vol. 6764, pp. 12–20. Springer, Heidelberg (2011)
2. Rouillard, J.: Developing a multimodal application for a scientific experiment on smartphone: case study, tools and results. In: Tools for Mobile Multimedia Programming and Development. Part of the Advances in Wireless Technologies and Telecommunication Book Series, pp. 68–80. IGI Global (2013)
3. Wakefield, J., Smith, D.: From Socrates to Satellites: iPad Learning in an Undergraduate Course. Creative Education 3, 643–648 (2012)
4. Parsons, D., Petrova, K., Ryu, H.: Mobile Gaming - A Serious Business! In: Proceedings of the IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education, WMUTE 2012, pp. 17–24. IEEE Computer Society, Washington, DC (2012)
5. Cortez, C., Nussbaum, M., Santelices, R., Rodriguez, P., Zurita, G., Correa, M., Cautivo, R.: Teaching science with mobile computer supported collaborative learning (MCSCL). In: Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education, pp. 67–74 (2004)
6. Susi, T., Johannesson, M., Backlund, P.: Serious games – An overview. School of Humanities and Informatics. University of Skövde, Sweden (2007)
7. Riedel, J.C.K.H., Hauge, J.B.: State of the art of serious games for business and industry. In: Proceedings of the 17th International Conference on Concurrent Enterprising, ICE 2011, pp. 1–8, 20–22 (2011)
8. Marfisi-Schottman, I., George, S., Tarpin-Bernard, F.: Tools and Methods for Efficiently Designing Serious Games. In: Proceedings of the 4th Europeen Conference on Games Based Learning, ECGBL, Copenhagen, Denmark, pp. 226–234 (2010)
9. Michael, D., Chen, S.: Serious games: games that educate train and inform. Thomson Course Technology, Boston (2006)

10. Herrington, J., Ron, O.: Critical characteristics of situated learning: Implications for the instructional design of multimedia, pp. 253–262 (1995)
11. Balme, L., Demeure, A., Barralon, N., Coutaz, J., Calvary, G.: CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces. In: Markopoulos, P., Eggen, B., Aarts, E., Crowley, J.L. (eds.) EUSAI 2004. LNCS, vol. 3295, pp. 291–302. Springer, Heidelberg (2004)
12. Vahdat, M., George, S., Serna, A.: Wizard of Oz in Designing a Collaborative Learning Serious Game on Tabletops. International Journal of Information and Education Technology 3(3), 325–329 (2013) ISSN 2010-3689
13. Delomier, F., David, B., Benazeth, C., Chalon, R.: Situated and colocated Learning Games. In: 6th European Conference on Games Based Learning (ECGBL), pp. 139–151. Academic Conferences and Publishing International Limited, Cork (2012) ISBN 978-1-62748-068-0, ISSN 2049-0992
14. Mariais, C., Michau, F., Pernin, J.-P.: The Use of Game Principles in the Design of Learning Role-Playing Game Scenarios. In: ECGBL 2010 Proceedings, Copenhague (2010)
15. Aldrich, C.: The Complete Guide to Simulations and Serious Games. Pfeiffer, USA (2009)