A Real Virtuality Application: The Real Farmer Game

Michail I. Tourlos, Aris I. N. Paraskevopoulos, Christos T. Pezirkianidis, Stavros S. Stavrianidis, Iakovos A. Pavlopoulos, George S. Tselikis, Nikolaos D. Tselikas and Anthony C. Boucouvalas
Department of Informatics and Telecommunications
University of Peloponnese
Tripoli, Greece
{tst09001, tst09012, tst09032, tst09041, tst07048, tselikis, ntsel, acb}@uop.gr

***Abstract*—The paper presents the concept of "Real Virtuality" and applies it to online games, by implying the real-time integration of real life conditions and elements into the virtual world of a game, rendering it as lifelike as possible. On this basis and by taking into account the real weather conditions in real-time, we designed a simple game prototype (i.e., the Real Farmer) based on the main concept of the popular game FarmVille. The concept, the architecture and the implementation issues of the Real Farmer game are all presented and analyzed. Finally, as a proof of concept and for measuring the efficiency and the reliability of the implemented algorithms that mix real and virtual environments in the game context, a comparison between the results in a real life farm and the corresponding ones in the game is presented, which validated our implementation.**

Keywords-Real Virtuality; Game development.

I. INTRODUCTION

For many people, computer as well as internet games' entertainment is part of their everyday life. Most of these games are trying to set the player into a fictional, and sometimes virtual, environment, following the virtual reality concept [1]. Thus, the game industry targets products with attractive virtual environments, in order to set the players even deeper in this virtual world, while, on the other hand, the virtual environment is sometimes preferred to be designed as realistic as possible [2].

The latter is following the "Real Virtuality" concept. The basic idea of "Real Virtuality" starts in 1991 from Mark Wiser when he published the article "The Computer for the 21st century", and describes that the "Virtuality" of computers, i.e., the ability to compute, view and alter data with a computer, will exist within our physical world, outside of an electronic shell [3]. In computer games, this could be achieved by implying the real time integration of real life conditions and elements into the virtual world of a game, in order to render it as lifelike as possible.

Having the above ideas in mind, we designed and developed a prototype online computer game, i.e., the "Real Farmer" game. The game is based on the main concept of the popular game FarmVille [4], that adopts the "Real Virtuality" concept, by taking into account in near real time the real weather conditions and using them in the gameplay.

The rest of the paper is organized as follows. Section 2 cites the related work on computer games that follow the "Real Virtuality" concept. Section 3 presents the concept of "Real Farmer" game. Section 4 presents system's architecture as well as the corresponding implementation issues. It analyses technical issues and explains the operation and the logic of its components. Section 5 tries to validate the reliability and the accuracy of the game, based on a real life experiment. The paper is summarized in Section 6.

II. RELATED WORK

There are several computer games trying to mix the real and the virtual world of the game. Ingress is a game developed by Google specially designed for Android mobile devices [5]. The players are divided into two factions and their ultimate target is to conquer the whole field of a specific geographical map of the real world. By using the GPS receivers of their mobile devices, the players are able to control and examine their nearby area and can interact with objects of this map, so, in most cases, the game requires physical presence in the corresponding area. The mobile client represents each player as a small triangle, surrounded by a circle area (20m radius), within which the interaction is possible through the corresponding game interface.

Lego has also developed a game for Android and iOS mobile devices, following the Real Virtuality concept. The game is called "Life of George" and the objective is to build specific constructions indicated by "George" (i.e., the mobile device), with real Lego bricks [6]. George indicates a construction that must be built quickly and accurately. Construction's rating depends on virtuosity. If a construction is a virtuosic one it will be ranked with a high rate; otherwise, with a low one. In order to get to the next level, each construction instructed by George has to be completed

 ISBN: 978-1-61208-292-9

within a specific time period. In this period, the player also has to use the camera of the mobile device, in order to capture a photo of the construction, to upload it through the corresponding interface and get to the next level of the game.

“Save 'Em” was designed to explore the challenge of making computer games more immersive [7]. Inspired by Lemmings [8], a classic computer game, Save ‘Em is based on maneuvering a group of slow-witted characters called Dudes through a treacherous maze. Using augmented reality techniques, Save ‘Em places virtual game entities directly within the player’s physical environment; gameplay takes place on a real game board rather than on a computer screen, and the Dudes’ fate is tied directly to the player’s physical actions.

There are also many games and platforms that try to feed the virtual world of the game with elements from real life, by using, in many cases, special joysticks or external control devices. Most of them are about real time “music”, “dancing” or “sports” games. For example, in “Dance Dance Revolution” by Konami, the player steps and dances on an appropriate arrow plastic mat to match the on screen arrow [9]. In “Guitar Hero” by Harmonix for Playstation 2 platforms, the player has to use a miniature of a Gibson guitar, while in “Donkey Konga” by Nintendo the player has to clap and drum in rhythm by using two small bongo drums [10]. Last, but not least, the Wii by Nintendo, was the first game platform that popularized new concepts like force-feedback in many games through the corresponding proprietary interface [11].

To conclude from all the above, we can see that the target of game industry is to design and produce more interactive games. Nevertheless, there are still very few games on the market based on the “Real Virtuality” concept. Probably the answer is that “Real Virtuality” needs more time to mature as a concept in order to be absorbed by the game industry. This is also another reason to present the “Real Farmer” game.

III. THE “REAL FARMER” GAME

The “Real Farmer” is a farming simulator online game, targeting to promote the most productive farmer. Through this game, the player has the opportunity to become a real farmer in a virtual world. Similar to every real farmer, the player has his/her own farmstead and he/she has to take care of it in order to achieve as bigger harvest as possible, and finally, get a corresponding profit by selling it. With this profit, he/she would be able to buy essential supplies such as fertilizers, seeds and farming equipment and use them to improve the productivity of the farm.

The first step of the player is to choose a region to build his/her own farm. By default, the proposed region is the one where the player lives, but there is also the opportunity to buy land for a second farm in another place, if he/she posses the required budget. The price of the land varies, depending on the corresponding fertility; more fertile land means more expensive to acquire. After this step, the game begins and involves the “Real Virtuality” concept in its own concept by applying in near real time the real weather conditions of the corresponding farm’s region, since the weather conditions differ from a region to another. In order to make the concept even more realistic, there is also a kind of help in the game providing useful agricultural information about the kind of vegetables or fruits that thrive in specific regions or climates. This information includes the proposed plants per region, some basic requirements for each plant (climate conditions, water, suitable fertilizers, etc.), the whole life cycle of the plant (in days), the cost of the seeds and the estimated profit from each plant (both in euros per kilo).

Based on all the above, the farmer takes a decision about his/her final virtual farming and up to this point the game follows the real life rhythm, with the mean that the actual time for the virtual plants growing up, the plants’ condition as well as the weather conditions in the farming will follow the corresponding real ones.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The Real Farmer is a network game designed for windows and mac operating systems. It follows the client-server model and requires an Internet connection.

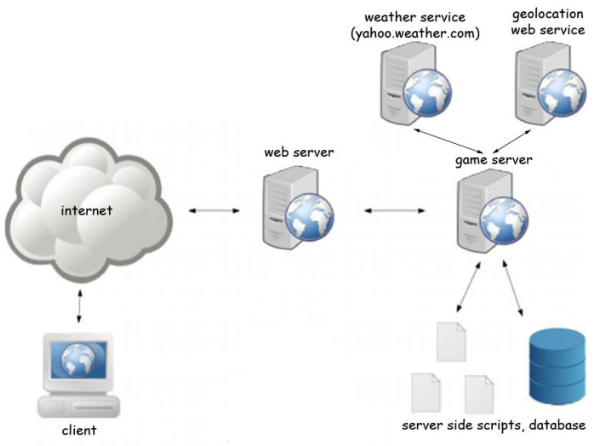

Figure 1. Real Farmer high-level architecture.

As shown in Figure 1, on the server side, there are two deamon applications, both developed in Java, residing in the game server. The first one is responsible to discover the location of a new player, by using a corresponding geolocation web service, while the second one is used to get the required weather information, from a reliable and hourly up to date weather forecast site [14]. Further up, the demon parses the data and stores it into the database as game parameters in an hourly rate. The above function is periodically triggered for every region where a subscribed user resides. Furthermore, the game server includes also the required scripts, which are handled via AJAX technique [15] by the client-side, in order to connect with, query and get response from the database.

The client-side is a graphical environment developed in Unity platform [12]. It illustrates the land with the vegetables or trees that the farmer decided to farm. The farming life-cycle is directly affected by weather conditions such as temperature, wind speed, rain emission, snow emission and cloudiness. These parameters alter in real-time (or, at least, in near real-time) based on the information stored into database in hourly rate. From a software perspective, both plants and weather conditions are Javascript classes that interact with each other. Actually, the state of the latter (i.e., the weather condition object) affects the corresponding methods of the former (i.e., the plants objects).

Following the Real Virtuality concept, the growth rate of the game's plants has to follow the corresponding one of a real plant. On the other hand, for the smoothness of a scene, game developers are usually updating game's variables per game frame. But in the Real Farmer case, the corresponding functions for updating a vegetable's growth is not required to be triggered every frame. Thus, they're periodically triggered in per second rate. Actually, this time interval is also short enough and makes the plant to appear stagnant, but we used it, in order to have a common basis in time scale (1 day = 24 hours = 1440 min = 86400 sec) in conjunction with the reality.

To be more specific, we chose one vegetable (i.e., the carrot) to describe the corresponding pseudo-code regarding plants' growth and how weather conditions may affect it, as described in Figure 2. Similar methods have also been defined for all supported plants of the game.

Essential weather parameters affecting carrots' health and growth rate are rain emission (a carrot needs about half liter of water per day), cloudiness (the farming must be ideally hit by sunlight about eight hours a day) and snow emission (snowfall destructs carrots). Wind speed is not affecting it at all, since carrot is actually a root and it's growing up into the ground. All vegetables support one main method for their growing, let us call this `plant_growing()`. They also include variables about their growth, growth rate, sun exposure, health and humidity. This is an abstract method, and it's specially modified for each plant available in Real Farmer game, rendering this way the game more realistic. If the farmer sows a number of seeds, an equal number of the corresponding plant objects are created (actually one object is created and the number of seeds is defined as constructor's parameter) and the corresponding attributes are set to fixed values, by default.

In the pseudo-code of Figure 2, `-15 * 10^(-5)` is the dehydration rate of the carrot, while `10^(-7) * rain_emission` is his hydration rate when it rains, and depends on rain emission.

```
growth = 0;
humidity = 50;
sun_exposure = 100;

plant_growing() triggered every second {
humidity = humidity -15 * e^(-5) + e^(-7) *
rain_emission;

if (sun is up AND cloud_cover_percentage < 100)
  sun_exposure = (sun_exposure + (100-
cloud_cover_percentage))/ 86400;
else
  sun_exposure = sun_exposure - 1/86400;

health = 2*humidity/3 + sun_exposure/3;
growth_rate = 1.5 * e^(-7) - 0.5 *(100-
health)*e^(-7);

if (health==0 OR humidity==0)
  object dies;

if (growth < 1)
  growth = growth + growth_rate;
else
  print("Harvest your carrots!");
}
```

Figure 2. Weather conditions affecting carrot's life (pseudo code)

If a player disconnects from the Real Farmer, all data concerning his/her game progress are uploaded into the server and are stored in the database. Subscribers' data will keep updating even if they are offline, because, as aforementioned, the plants in the Real Farmer grow up in real time. When a player resumes the game, all previously saved information, as well as the time he/she was offline, will be sent back to his client. The game's plants never stop growing up, unless they die or if they are ready to be harvested. To fill the gap about what happened when a user was offline, a feature has been implemented to play in fast forward a mini clip of the farming evolution or destruction. The fast forward factor is equal to 3600/5 (i.e., one real time hour is equally simulated to five seconds in the mini clip).

V. REAL FARMER VALIDATION

In order to validate the simulation activities that take place in Real Farmer, we had only one alternative, i.e., to be real farmers ourselves. This way, we can demonstrate not only the validation of the graphic representation of weather conditions (this is the easy part), but also the reliability and accuracy of the algorithms used for plants' growth.

For the first test, the left part of both Figure 3 and Figure 4 depicts captured photos from Tripoli, Greece, during a sunny and a cloudy day, respectively. In the same figures, at the right, you can also see the corresponding screenshots taken from the Real Farmer at the same date and time. It is obvious that a satisfactory depiction of the real weather has been achieved. On the other hand, since the Real Farmer depends on an internet weather forecast and not in real time weather data, the accuracy of weather depiction depends on the accuracy of our internet data source. Even if there are several internet weather forecasts which are more reliable for the weather conditions in specific countries (e.g., [14] for Greece), we preferred to use Yahoo weather forecast and render Real Farmer maybe less reliable in that sense, but applicable to any region worldwide, on the other. Regarding the delay on depiction of changing weather conditions, the mean delay time is about 3 sec. This is actually the required time for the weather update in data base entries by the weather forecast service plus the time needed for re-initialization of weather parameters in the game. The delay of 3 sec can be safely characterized as negligible compared to the time that the respective weather conditions are depicted in the game (i.e., 1 hour, since the weather is updated in hourly rate).

Figure 3. A sunny day (reality vs. Real Farmer)

Figure 4. A cloudy day (reality vs. Real Farmer)

For the second test, as aforementioned, we had to be real farmers by ourselves in Tripoli, Greece. So, we sowed 10 seeds of carrot at the University of Peloponnese campus and we started the Real Farmer game in parallel. The experiment lasted for three months (November 2012 to January 2013) and the results were encouraging enough, since, as depicted

 ISBN: 978-1-61208-292-9

in Figure 5, in 88 days we were ready to harvest the real carrots, while in Real Farmer we needed only 4 more days, 92 in total.

Some additional information that is not mentioned in Figure 5 is that during the 92 days that the experiment lasted, we observed 23 sunny, 40 cloudy and 29 rainy days.

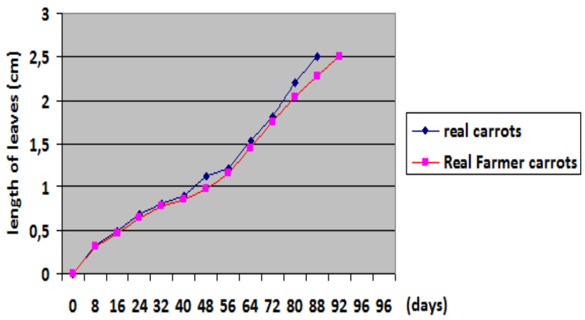

Figure 5. The carrot life cycle (reality vs. Real Farmer)

We totally calculated 94mm of rainfall and 0mm of snowfall. The mean temperature was 10.5^{o} C and the mean humidity was 76,8%. The wind speed does not affect carrots' life at all; thus, we didn't calculate it. Last, but not least, the carrot's growth is associated with the length of its leaves, since carrots are ready to harvest if the mean length of their leaves is 2.5 cm.

VI. CONCLUSION AND FUTURE WORK

Based on the "Real Virtuality" concept, we presented and analyzed the concept, the architecture and the implementation issues regarding the "Real Farmer", a simple game prototype. We also described the experiment that took place and we presented the corresponding results in order to evaluate its reliability and accuracy and validate the implemented algorithms.

In order to make "Real Farmer" more realistic, we are currently working on fine-tuning of the growth algorithm regarding most of the supported plants. Furthermore, we also plan to give a business-oriented perspective in the game, by trying to bind the final prices of the fruits and vegetables to the real market ones, at least for European and North American counties where this info is available. Finally, following the current trend, we are about to deploy the "Real Farmer" for Android mobile devices, expecting more registrations and, thus, more feedback to improve our work.

REFERENCES

[1] G. Burdea and P. Coffet, "Virtual Reality Technology", Second Edition, John Wiley & Sons, 2003.

[2] P. Zackariasson and T. L. Wilson, "The Video Game Industry: Formation, Present State, and Future", New York: Routledge, 2012.

[3] M. Weiser, "The computer for the 21st Century," IEEE Pervasive Computing, vol. 1, no. 1, 2002, pp. 19-25.

[4] FarmVille, online: http://en.wikipedia.org/wiki/FarmVille, [retrieved: July, 2013].

[5] Ingress,online: http://en.wikipedia.org/wiki/Ingress_(game), [retrieved: July, 2013].

[6] Life of George, online: http://george.lego.com/en-us/, [retrieved: July, 2013].

[7] C. Watts and E. Sharlin, "Save 'Em: physical gameplay using augmented reality techniques", ACM conference on Future Play, 2007, pp. 160-165.

[8] Lemmings, online: http://en.wikipedia.org/wiki/Lemmings_(video_game), [retrieved: July, 2013].

[9] Dance Dance Revolution, online: http://en.wikipedia.org/wiki/Dance_Dance_Revolution, [retrieved: July, 2013].

[10] Donkey Konga, online: http://en.wikipedia.org/wiki/Donkey_Konga, [retrieved: July, 2013].

[11] Nintendo-Wii, online: http://www.nintendo.com/wii, [retrieved: July, 2013].

[12] Unity online: http://unity3d.com/, [retrieved: July, 2013].

[13] Yahoo weather online: http://yahoo.weather.com, [retrieved: July, 2013].

[14] Meteo weather online: http://www.meteo.gr, [retrieved: July, 2013].

[15] C. Draganova, "Asynchronous JavaScript Technology and XML (AJAX)", 11th Annual Conference of Java in the Computing Curriculum, 2007, London.