

Kroster-MHP Game for Digital TV. Developing Process, Design, and Programming Considerations Against Technical Issues

Iván Abadía Quintero, Madelayne Morales Rodríguez, *Student Member, IEEE*, Camilo Ortegón Barajas, Juan Vicente Pradilla Cerón, Patricia Madriñán Rodríguez, and Andrés Navarro Cadavid, *Senior Member, IEEE*

Abstract—This paper presents the development of Kroster, a serious game created for digital television using multimedia home platform (MHP) technology. The process of game creation is described from both an engineering and design point of view with discussion on various programming and graphical aspects. Some of the requirements for the development of a t-learning game are described, as are the restrictions associated with the TV platform, especially within the MHP development framework.

Index Terms—Digital television, game, t-learning, multimedia home platform (MHP), design and programming, technical aspects.

I. INTRODUCTION

DIGITAL television allows access to distant regions that have poor communications infrastructure, but rich diversity. However, consciousness regarding this diversity is not always present [23]. Typically, terrestrial TV content deals with entertainment and not education, because to educate without interaction is very difficult or impossible.

The deployment of digital TV in Colombia in 2009, using the European DVB-T standard and, more recently, the DVB-T2 standard in 2011, has presented different opportunities for the development of interactive digital content intended to reach distant areas of the country. The t-learning concept using the serious game approach is an interesting opportunity for a country with diverse geography with areas of low population density, in which radio and television are the only available communication technologies.

The adoption of European standards has allowed the development of interactive applications by means of Multimedia Home Platform (MHP) technology, in conjunction with t-learning experiences using both Jinga and MHP or other proprietary tools. This has provided the motivation for this work, which is based on MHP and the serious game paradigm. The main concept behind the game is environmental responsibility

Manuscript received July 11, 2013; revised October 2, 2013; accepted October 2, 2013. Date of publication October 16, 2013; date of current version November 26, 2013. This work was supported by the Colombian Departamento Administrativo Nacional de Ciencia, Tecnología e Innovación-Colciencias through Grants ARTICA and Succes-TV under Contract 087-2012.

The authors are with the Laboratorio de Investigación en Informática y Telecomunicaciones i2t, Universidad Icesi, Cali 122-135, Colombia (e-mail: made1989@gmail.com; iaabadia@icesi.edu.co; camilo.ortegon@correo.icesi.edu.co; juan.vicente.pradilla@gmail.com; patri@elsitiomio.com; anavarro@icesi.edu.co).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/RITA.2013.2284961

and the biodiversity conservation of endangered species in the country.

Creating a game for digital TV is focused on two main aspects: design and programming. Each area has an intrinsic relation with the technical aspects of TV and with the “playability” aspects of the game. An objective of both areas must be to perform with the media, which requires the determination of the main obstacles and a trial and error process. The Kroster development process was iterative, developing parts of the game and adjusting various aspects in both the design and the programming as became necessary during the development process. Both aspects were developed in an independent manner with feedback during the simulation and testing stages.

The general design process is shown in Fig 1. This paper is organized as follows. In section II, we show the motivation and antecedents of the work. Section III presents a description of the game and describes both the design and the programming considerations, and discusses the graphical and technical restrictions of the MHP-enabled receivers. In section IV, the programming process is described and so are the restrictions imposed on the game development by the limitations of the MHP and TV systems discussed. Section V presents the graphical design aspects and decisions taken during the development process to achieve a “good” graphical interface and the desired “playability”. Finally, section VI offers conclusions and thoughts on future work.

II. ANTECEDENTS AND MOTIVATION

Serious games have grown in importance in recent years. The main antecedents have come since the publication in 1970 of Clark Abt’s book *Serious Games* [1], and have derived new momentum in the last five years thanks to the development of 3D game technology and faster graphics cards.

For interactive TV applications, available technologies such as MHP or Jinga, as well as the computational capacity of set top boxes, have imposed important restrictions on the development and use of the serious game concept in DTV. The following paragraphs present a short revision of the serious games and interactive TV games that motivated this work.

A. About Serious Games

A serious game is one in which the main goal is oriented towards information and learning, using entertainment as the main communication tool [7]. Generally, there are many

Simulation and testing (Xlet/STB)	Development of slides handler code	Adjustment of layer's code base (perspective)	Adjustment of layer's container 3D code base	Development of sprites handle code	Graphic's adjustment of 1rs level + map + bonus	Development and integration of the 2nd level	Simulation and test (Xlet/STB)	Adjustment of current code base for the 3rd level		Development and integration help frames + 3rd level	Simulation and test (Xlet/STB)
	Development of slides for animation	Image size and proportion adjustment	Development of contextual menus	2nd level prototype ("paramo")				Development of help and game introduction frames	3rd level prototype ("whales rescuing")		

Fig. 1. Kroster game creation process.

different opinions about whether serious games are related directly to educational games [5], [6], or whether they must be related to informational and persuasive objectives oriented to training [3]. These kinds of applications have user interaction characteristics where, by means of fun, entertainment, games, and ludic activities, some transfer of information is achieved [4]. Many different types of serious games exist, from simple games (e.g., card memory games and trivia) to complex games that simulate alternate realities (e.g., simulation games, augmented reality, and immersion games) [2]. In addition, there exist other classifications related to the area of practice, such as educational, health, military, transport, and multimedia [3].

B. In Relation to Television

Although there is no direct relationship between serious games and television, these two areas have been defined to promote educative entertainment through TV interaction. On the one hand there is t-learning, which is a convergence of technologies that promote learning through television [8]. This is an area of constant growth, which ensures that learning through TV interaction is emerging as a key mechanism through which to acquire and develop knowledge, skills, and to encourage communication, analysis, and discovery [9]. On the other hand, there is edutainment (a word derived from the combination of education and entertainment), which is a concept that refers to television segments that complement learning elements [10], making the learning experience into something more fun through challenges and entertainment activities based on [10] and [11].

In general, through educative entertainment via the television, the user/gamer develops their own knowledge starting from the contact with the knowledge, and using several tools: learning by doing, learning through mistakes, learning by contact or interaction, role-playing games, and constructivist knowledge. The contents of the apps and games may have goals, rules, and different competitive modes, in addition to generating feelings, sensations, and opportunities to motivate the user/gamer to play, learn, and assimilate certain types of information [12].

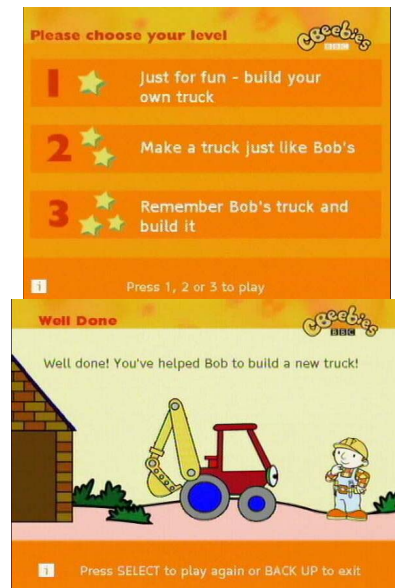


Fig. 2. Build a truck Application [17].

C. Some Antecedents of TV Games

A review of several different educational apps for digital television exposes the underlying concepts.

1) *CBeebies*: Developed by BBC (UK) channel, this is a channel for children under six-years old. Its principal scope is to develop and strengthen the children's skills in preschool studies, as shown in Fig. 2. This channel broadcasts audiovisual content and different programs, mixed with interactive content (e.g., games, trivia, karaoke, etc.) related to the transmissions [17].¹

2) *A Turma Da Árvore*: This is a program on Brazilian television with a large audience, developed for children up to six-years old with analphabetism issues, shown in Fig. 3. It was a pioneer in Latin American television in including interactivity in a television broadcast. The areas covered in this program are environmental education, health, and citizenships

¹<http://www.bbc.co.uk/cbeebies/> is the channel web version of this app.



Fig. 3. Game called “Nem todo lixo é lixo” from the program A turma da Árvore. Here, the children have classified the garbage into different recycling groups [16].



Fig. 5. Active learning graphical user interface [14].

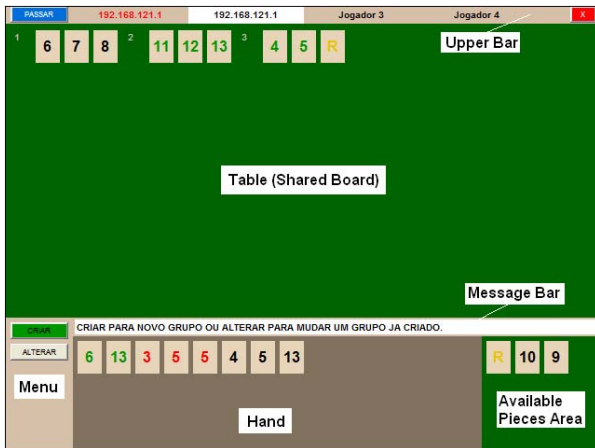


Fig. 4. Rummikub graphical user interface [15].

issues, through the use of puppets that live in a tree-house. The interactive version of this program provides different communication channels between the student and the teacher, and presents interactive games that help tell the story, mixed with break points decided on by the user [16].

3) *Rummikub* : This is a mathematical game from the RummiTV channel, the main objective of which is to stimulate math and logical reasoning skills in children and young people over eight-years of age. The game consists of a series of tiles placed on a board, where the player must compete with other players to get rid of all the tiles they have. The method of getting rid of them is by forming groups of records and performing a series of operations with them (by color or numbers) [15]. Shown in Fig. 4.

4) *Active Learning*: This is an interactive module developed by the company Tata Sky DTH-Hindu, which is aimed at children between the ages of 7 and 12 (Fig. 5). It is part of a series of interactive television applications ranging from learning English to cooking, games, and music. This module is presented in the form of quizzes in several topics, such as mathematics, science, food, etc., where the child can respond and accumulate points with which they can claim prizes [14].

D. The Motivation Behind Kroster

Countries in regions outside of the tropical zone view their part of the world as having four seasons, whereas an equatorial country such as Colombia bases its climatic differentiation on

altitude. The same climate persists all year depending on the height above the sea. For someone used to the four seasons, it can seem very odd that the climate of a country is spatially and not temporally dependent. Animals and plants depend on a particular habitat at a particular location and this is how it works in this country of timeless physical contrasts.

The phenomena of thermal levels and the diversity of animals living at different altitudes, from sea level to the snow line above five thousand feet, motivated the development of a game that allows the player to learn in a pleasant way about the fauna in these different climatic zones.

This is how the proposed “Kroster” game was born. The name derives from a ride “kros” (cross) through “ther”mal (“tér”ricos) levels, on three levels that allow the user, viewer, and player, three in one, to make the journey changing places from the first person point of view, to the third person from an overhead or elevated view.

III. GAME DESCRIPTION AND DESIGN CONSIDERATIONS

A. Design Considerations

Kroster is a game whose main theme is to develop some consciousness about biodiversity conservation. Its main learning goal is the logical association between Thermal Floors, climate conditions, and existing ecosystems. To reach the goal, we use information layers and schematic images to communicate to the player the main characteristics of a Thermal Floor.

For this process, the change in the point of view is very important. For a person born and raised in one place, their world is their immediate surroundings and thus, they have nothing with which to compare when evaluating its importance.

Kroster is a game structured organically: initially the route is climbing and then sliding, and finally, it is set in a backwater, using an analogy of the lifecycle.

The Beginning is in the first person at sea level. Then there is a mountain climb, initially viewed from afar, allowing the player to become conscious of their surroundings.

The Climax of the game is reached at the top of the mountain, where the scenario changes and the player view is in first person moving down a river.

In the first part of the game, movement is restricted; movements are only vertical while the bicycle is moving forwards.

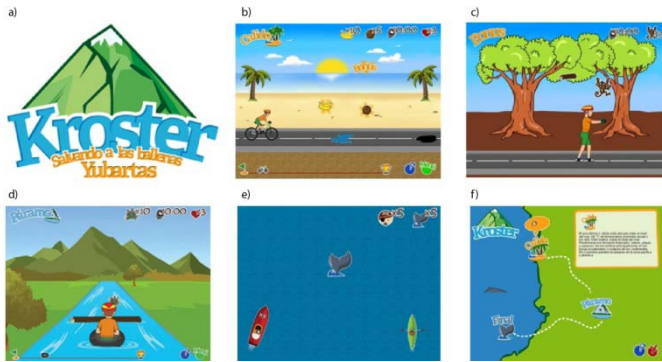


Fig. 6. a) Game identity - logo; b) Warm Thermal Floor - Level 1; c) Bonus level 1; d) Paramo Thermal Floor - Level 2; e) Final level. Race against pirates; and f) Initial map with all level access.

The user must jump over obstacles using the up arrow. During the descent, the movements are horizontal and the obstacles must be avoided by using the left and right arrows.

1) *Game Denouement*: In the final part of the game, the player reaches the sea and the point of view changes to the zenith (Fig. 6e). Now the player has a complete view, in which both horizontal and vertical movements are possible at the same time, enriching the games learning experience.

B. Game Description

Kroster has three levels, consistent with the three points of view of the player, which have to be completed within a limited time. The player wins points according to the items collected, the time spent, and on reaching the level goal: the rescue of endangered species.

In the first level, the player has the role of a BMX player (Fig. 6b). The view is in third person crossing the warm Thermal Floor. During their passage, the player must avoid different obstacles (such as tires, holes, and oil spots) and collect a number of autochthonous fruits from this ecosystem. To complete the level, the player must rescue the Spider monkey. In this level, a bonus is issued if the player is able to avoid falling trees and catch additional monkeys (Fig. 6c).

The second level (Fig. 6d) shows the player rafting in the cold (paramo, above 3000 meters) Thermal Floor with a first person (behind the player) point of view. The player should avoid trees, chops in the river, and collect friarys, a typical paramo bush. To finalize the level, the player should rescue a Tapir, an endemic endangered species.

The third and final level (Fig. 6e) uses a zenith view where the player is the commander of a boat. The object is to rescue the highest number of humpback whales, competing against the whale-hunting boats that are trying to capture them.

Each of the levels can be accessed by means of a map. This map provides the player with relevant information about each Thermal Floor and the score is shown (Fig. 6f).

IV. PROGRAMMING PROCESS CONSIDERATIONS OF TECHNICAL ASPECTS AND DEVELOPMENT CONSTRAINTS

MHP [24] is middleware developed by the DVB forum and standardized by ETSI for the development and execution of

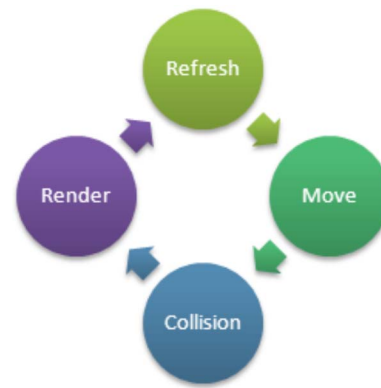


Fig. 7. Basic game cycle.

interactive applications in the European digital TV framework, which is intended to offer a common platform for applications interoperability irrespective of the manufacturer of the set top box or TV appliance.

Because of the low computational capacity of set top boxes, MHP developers placed restrictions on graphical management and memory. This is why game development for MHP differs considerably from the development of games for computers or consoles.

In practice, the deployment and availability of MHP-capable devices has been quite restricted, which imposes additional restrictions on projects such as Kroster. Additionally, the lack of developer friendly MHP tools restricts the possibility of non-software skilled designers developing MHP applications.

In order to deal with such restrictions, the development team has developed the TVGame library, designed to simplify game development on MHP platforms. The TVGame library is a library for the development of MHP games, carefully designed and built during the development process of Kroster. The library consists of a set of Java classes oriented to simplify the development of games, which includes the management of sprites and animations (sprites). It is a starting point for the development of MHP-based applications and it is based on the basic cycle process of a game (Fig. 7), on the application cycle, and on the required methods that should be defined for the game cycle (Fig. 8).

A. Programming Aspects

The development process of Kroster was undertaken using the TVGame library, which simplifies the management of images, layers, collisions, and animations during the game development.

As mentioned previously, the main idea behind the development of the TVGame library was the development of an independent library for MHP game development and a .jar version for distribution. Although Kroster was not the first application developed using this library, it is the first complete game that has been developed using the library. During the development process, some modifications to the TVGame library were identified in order to make it more useful and usable, but retaining the original intention of the library.

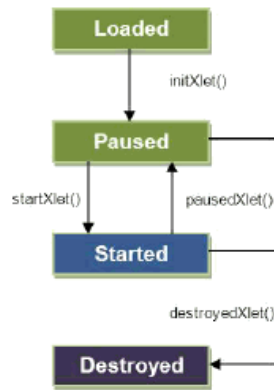


Fig. 8. Application cycle.



Fig. 9. Perspective effect in the game. Although the trampoline is aligned with the palm tree, the layer manager allows its superimposition, depending on the player's position.

The TVGame library comprises a simple, generic, and light structure with basic classes for layers, images, and canvas management (i.e. *Layer.java*, *ImageLayer.java*, *Road-Layer.java*, *Sprite.java*, *TiledLayer.java*), graphical resources load (*Resource.java* *Layer.java*), and the main game class (Fig. 7) (*Game.java*).

In Kroster, the classes *Layer.java* and *LayerManager.java* were developed for the layer management in each game level (e.g., background, main character, and the interacting objects). This gave freedom to the programmer to focus on the actions and movements in the game scenario, instead of placing attention on the synchronization of different elements and their locations on the screen, as demanded by the MHP standards. Because of the player's point of view and because of the characters interaction with background elements, objects in the game space appear differently depending on how far they are from the camera in each level (because of perspective). Therefore, it was necessary to develop a class *PerspectiveUtil.java* in the TVGame library. This library allows a 3D worldview, as shown in Fig. 9.

Initially, it was reasonable to use the *PerspectiveUtil.java* class to give a realistic effect to the scene, keeping the existing restriction of 15 simultaneous layers. However, in the first level of the game, this limit was exceeded. The level had more than one hundred elements in the scene, complicating the use of the class, and the synchronization of the movement and the location of the different elements on the screen. Because of this, the task was delegated to the layer manager, giving additional capacity to the programmer to control the game automatically (the management of the perspective effect was assigned to the *Layer Manager*) or manually (considering the

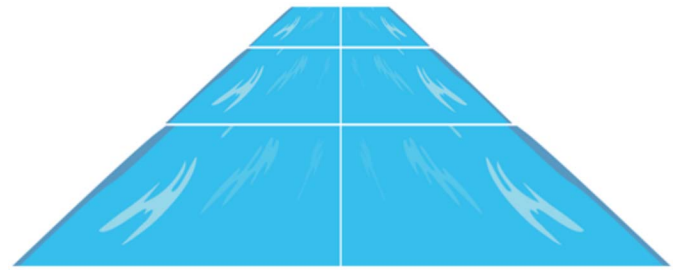


Fig. 10. Road-type image. River built in level 2.

necessary adjustments to the positions of the objects in the different layers). However, even after this additional adjustment and library modification for perspective management and layer synchronization, this was still insufficient for the complexity of the second level, where a first person 3D effect is used, and where objects are approaching the players as they descend in the river.

There were two alternatives for dealing with the changing size of the objects, according to their proximity to the character: the first was to overload the STB processor modifying the image size in real time, in order to obtain an array of different sized images and to paint it dynamically according to the position in the screen. The second option, which was the most efficient solution found by the team, consisted of the use of five preloaded sizes of the same image. Memory loading with different image sizes is a very delicate operation, because it can make the application crash. For the execution of the preloaded image process, the *Resource.java* class in the TVGame library was used to import the images. This class allows an image to be imported once, such that if the image is requested, the image is the same as previously loaded, optimizing the STB resource usage. However, if the previously loaded image is used to create a copy with modified characteristics (such as the size), the initial reference is lost and the original image changes. This means that anytime the size is changed using the *resize()* method from the class *Resource.java*, a new object is obtained with different memory space. Therefore, it is necessary to ensure that the size change operation is made one time only and that the image is used on each object that needs that image size.

For the adequate visualization on the river with the perspective effect, it was necessary to consider that the road-type images are repetitive, both in depth and symmetry, in such way that it is not necessary to store the full image, but only small pieces that are assembled dynamically. An example is shown in Fig. 10.

The *sprite.java* class is responsible for the handling and operation of *sprites*. Sprites are game layers used for element animations, whose shapes can change at any moment. The utility of the sprite is the simple use that allows segmenting the image in small parts, which provides the sensation of movement. Additionally, such animations can be configured in such a way as to show only the movement of some frames (e.g., to move a leg).

In the second level, the objects had to appear from the horizon, sailing in the river, and finally pass the character

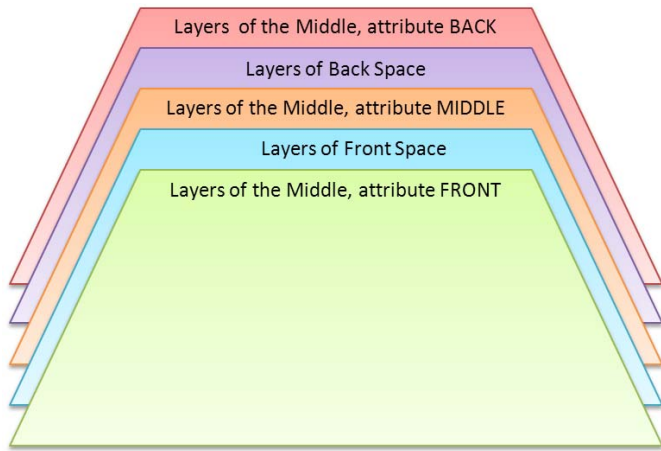


Fig. 11. Layer prioritization in level 2.

surfing on the floater; thus, the layer management used in the first level had to be revised. For this problem, a new layer container class was developed with similar functions to those existing in the *Layer Manager*, but with the possibility of managing three different spaces (front, middle, and bottom). The layer type class *SpacialContainer.java* was developed, which was added to the *Layer Manager*, and was responsible for the layer painting process on each level.

In order to obtain the depth effect and the image resizing according to the travel path, we determined that all the layers stored in the front space and the bottom never change in priority at the moment of painting (e.g., the character floating in the river). Furthermore, the objects located in the middle space do not have a fixed priority, and change according to the relative position attribute, taking three possible values: *back, middle, or front*. Priority is shown in Fig. 11.

The *SpacialContainer* class, developed especially for Kroster, was not included in the TVGame library, because it uses objects not conceived for a game library (those with relative position and Z-axis). Moreover, a position estimation function was implemented for the X- and Y-axes, according to the parameters shown in Fig. 12.

- Vision elevation (A): measure used to determine the inclination of the projected ground and to estimate the required movement;
- Approach percentage (B): percentage of distance from the horizon to the character (C). This percentage can take negative values because of objects passing the character;
- Y-axis target (D): location in Y-axis where the main character is located waiting to be reached by surrounding objects;
- X-axis center (F): axis dividing the screen into two parts, left and right;
- Origin point (G): X and Y coordinates where the object departs; and
- Relative dimensions (H): width and height of the object at a specific instant.

The function mentioned previously is implemented in *DimensionalDealer.java*, but the estimation of the approaching percentage is done in the *Objeto3D.java* class. As those classes



Fig. 12. Parameters used to determine an element position in the screen.

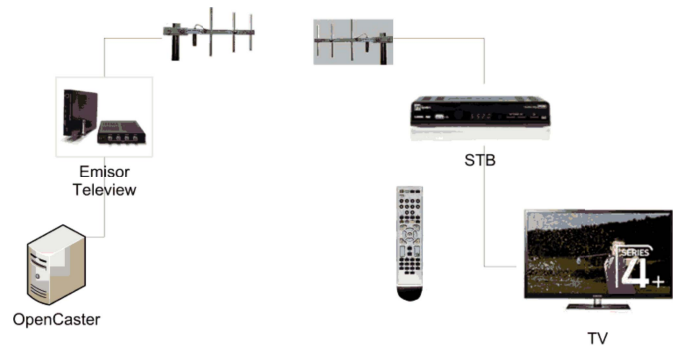


Fig. 13. Lab test bed used for Kroster.

do not belong to the TVGame library, this does not require any specific package order or to be generic. Therefore, it can be adapted to the specific requirements of this game. Collisions between the main character and river objects are in the XY-plane, and then it was necessary to create a depth attribute.

Even though the game library allows the redefinition of collisions as advanced collisions, it was not possible to adapt this function to the second level of the game. This was because the library has to check all the absolute collisions in the XY-plane before redefining a collision. Because the river collisions are in the XY-plane, the library has problems with the management of both functionalities.

B. Test and validation

Because of the restrictions of the MHP system and because the STB are not general-purpose computers, the development of the game was done on a conventional computer. Before loading the game on the STB, it must be simulated in a computer with an MHP emulator, and then tested with an MHP-compliant STB connected to a TV.

The simulation was done using an emulator that executes Xlets,² which provides an idea about the performance and

²Xlet is the given name to software components in MHP, of similar form to an applet in Java.



Fig. 14. Example of planning and sketching of the game.

behavior of the game in a real STB. For Kroster, we used XleTView [22], whose configuration and operation depends on a file developed by our engineering team, and is not distributed with the original XleTView software. The simulation process consists simply of executing the game and testing its functionalities, interface, and usability in a similar way to a real STB. In the case of Kroster, computer tests were done leaving aside some visual functions available in the simulator, but not available in MHP. Such functions (from the *Resource.java* class) were not in the library and were not used for Kroster development.

For the tests in the STB, it was necessary to emulate the real environment of a DVB-T/T2 system. For these tests, a DTV lab available in our research group was used. This consists of a Televue Lab transmitter with DVB modulations, which converts a DVB-ASI/SMPTE-310 MPEG video stream from a computer into a modulated air signal in the 55–956 MHz and 956–2150 MHz frequency range; a Telesystem MHP-enabled STB; and two antennas, one for the transmitter, one for the receiver. In order to produce the interactive content and to incorporate the MHP stream into the ASI stream, OpenCaster was used [20], which is a free software system designed to generate, process, multiplex, and transmit interactive applications over an MPEG-2 TransportStream broadcast according to the DVB standards. In order to transmit the application, it is necessary to have the source code, images, and *.class* generated when the application is developed, and to transmit it from the Televue to the STB. The lab test bed is shown in Fig. 13.

V. DESIGN PROCESS. CONSIDERATIONS AGAINST TECHNICAL ASPECTS

The development process of a digital TV application is constrained by several technical factors that affect the way the application is developed and conceived from the creative point of view for DTV requirements [18]. For the specific case of Kroster, the first design step was to sketch on paper its functionalities, general structure (in a sketch) and, at the same time, its playability. This planning process is shown in Fig. 14.

Additionally, the game identity was established (logo and color palette) and the iconography, interaction aspects, and typography selected. Technical restrictions related to the size of the images limited the construction of a graphical proposal

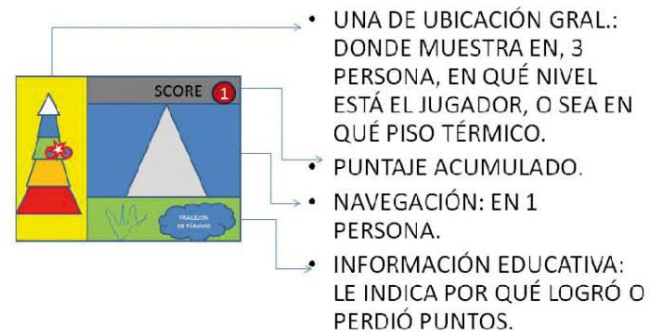


Fig. 15. Contextual elements and their appearance to improve readability.



Fig. 16. Typography used in the informative elements.

and had an important impact on the definition of the graphical aspects.

This restriction limited the use of diverse color palettes and of other effects, such as shades and degradés. Because of this, images are schematic and have high contrast level, especially for such images where a good readability was required for good interaction, which were delineated with a white edge (Fig. 15).

Regarding the typography, despite the MHP recommendation of the use of typography for television applications [19], [21] (such as Helvetica, Arial, or Verdana), the available STBs do not allow the use of fonts different to those pre-installed by the manufacturer, limiting the use of fonts described in the literature and MHP documents. It was then necessary to develop some contextual elements as images, as those shown in Fig. 16, assuring the quality of the information and the graphical consistency throughout the game. For lengthier texts (such as information areas), and given the restriction of typographical fonts, Tiresias was used, because it was the only one available in both STBs.

The definition of user interaction was limited by the input devices available—the remote control in our case. The user can manipulate the movement of the character in each of the four levels by using the control arrows. Additionally, the colored

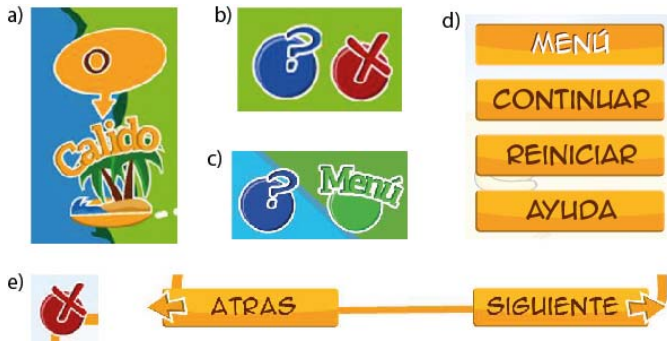


Fig. 17. Game interaction parameters. (a) main map; (b) main map, blue key to obtain help and red key to exit the game; (c) inside a level, blue key to obtain help and green key to show the options menu; (d) inside the menu, to browse options, up and down arrows, and to choose the option, the enter key, and (e) in the contextual windows for help and instructions, red key to close the window and left and right keys to browse between slides.



Fig. 18. Image definition problems using PNG-8.

buttons and the *enter* key were used to control playability options (menu, game help, exit, choose a level). In Fig. 17, interaction parameters for the main aspects of the game are shown: a) main map, to choose the level by using the arrows and *enter* key, b) main map, blue key to obtain help and red key to exit the game; c) inside a level, blue key to obtain help and green key to show the options menu; d) inside the menu, to browse options, up and down arrows, and to choose the option, the *enter* key, and e) in the contextual windows for help and instructions, red key to close the window and left and right keys to browse between slides.

When the first level prototype was developed, the main elements were set in two depth levels, trying to emulate spatiality and the sensation of travelling. The existence of a bottom wider than the screen width, which is moved slowly, and the use of many perspective levels with static and dynamic elements (e.g., moving clouds) were used in order to obtain depth perception.

Most images were exported in PNG format (Portable Network Graphics), because of its transparency. Others, such as the background and the road, were JPEGs, in order to maintain small file sizes. In order to obtain the smaller image sizes, and considering that the color palettes were restricted, we opted to use the PNG-8 format in a web-optimized scheme, reducing the number of colors used (between 6 and 8). After the development of the first level of the game and the simulation in the XleTView, some problems were detected such as edge definition, and size and proportion problems, due to the export process (Fig. 18). Therefore, the export format was changed from PNG-8 to PNG-24. This increases the size, but guarantees better visualization.

Moreover, because the resolution used was determined by the screen resolution (1024 × 768), commonly used in most computers, some problems arose when we tested it on a TV (Fig. 19). Consequently, adjustments to the proportions and elements on the screen were made, based on a similar



Fig. 19. Proportional differences and element locations between the original concept (a) and the XleTView emulation (b).

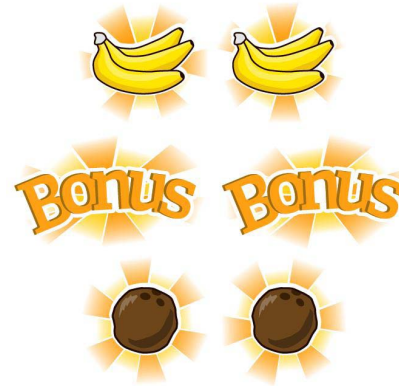


Fig. 20. Animated sprites for relevant elements in the game.

proportion near 800 × 600 px. This decision was based on the following reasons:

- XleTView does not have the option to emulate different screen sizes (the default configuration is 800 × 600);
- it is not possible to emulate the different scale between a conventional computer screen (4:3) and a TV screen (16:9);
- graphical tools or conversion utilities to convert images directly to TV, by converting one computer pixel (1 × 1) to a TV pixel (1,067 × 1 computer pixel), do not exist.

Although the first level had many depth levels, the user could not differentiate those elements that were part of the game from the decorative elements. Therefore, the relevant elements, i.e. those that the player picks during the game, were identified by an animated luminous halo (Fig. 20).

Moreover, to animate objects, we use sprites or successive images that simulate the movement from image to image (Fig. 21). This image series is built with a minimum set of images to guarantee small files, but with an adequate quantity to allow fluid movement on the screen. In many cases, especially in levels 2 and 3, some problems appeared in finding the correct number of required sprites. Furthermore, it was necessary to add a white sprite at the beginning of each sequence to visualize the animation correctly.

The design process was completed with the development and programming of the prototypes of levels 2 and 3 with the defined parameters, once level one was operating successfully.

On each level of integration to the main application, adjustments and corrections were made, especially in changes to images and graphical details, the necessity for which was observed during the simulation and user tests.



Fig. 21. Sprites example – cyclist level 1.

Finally, we established that the addition of functionalities to the navigation process (e.g., to enter using a color key instead of the *enter* key) was a detriment to the usability and user friendliness, because of the excessive load of additional information. Therefore, it was necessary to make important changes to the way each level was accessed and to obtain information about it.

VI. CONCLUSION

The use of MHP as a development tool for games has represented a challenge in the development of classes for the management of game logic and game views. In some cases, it was necessary to use old computer game techniques in order to accommodate the MHP restrictions.

To determine the management of the perspective of the game is very important, and despite the existence of classical views and perspectives, it is very important for the programmer to decide the views in order to simplify the development process and the graphical deployment.

Regarding the design, it is important to mention that although some recommendations about graphical and technical aspects exist for application development, no technical rules exist, which allowed the design team to establish the creative rules. This means that the design process must be iterative, adjusting the graphical issues. Despite this, the building of Kroster has facilitated the understanding and development of some design recommendations. From the design side, it is important to have studies on the design recommendations for DTV development, considering new technologies such as HbbTV and CSSTV.

Despite the graphical and code resource optimization on the STB side, it is important to improve the broadcaster side, by using optimizing tools like OpenCaster to achieve efficient transmission of the interactive content.

The development of interactive TV games is a multidisciplinary process combining technical, design, and usability skills, as well as pedagogical concepts related to the media.

As the logical evolution of Kroster, we are developing an HbbTV version that will be useful in comparing the technical and design aspects.

REFERENCES

- [1] C. Abt, *Serious Games Lanham*. Lanham, MD, USA: University Press of America, 2002.
- [2] C. Aldrich, *The Complete Guide to Simulations and Serious Games*. San Francisco, CA, USA: Pfeiffer, 2009.
- [3] D. Michael and S. Chen, *Serious Games. Games that Educate, Train and Inform*. Mason, OH, USA: Course Technology, 2006.

- [4] M. Ulicsak. (2010). *Games in Education: Serious Games [Review for Futurelab]*, Futurelab, Bristol, U.K. [Online]. Available: http://media.futurelab.org.uk/resources/documents/lit_reviews/Serious-Games_Review.pdf
- [5] A. Derryberry. (2007). *Serious Games: Online Games for Learning*, Adobe, San José, CA, USA [Online]. Available: http://www.adobe.com/resources/elearning/pdfs/serious_games_wp.pdf
- [6] B. Holm and B. Meyer, "Serious games in language learning and teaching—A theoretical perspective," in *Proc. Digit. Games Res. Assoc. Conf.*, 2007, pp. 559–566.
- [7] T. Susi, M. Johannesson, and P. Backlund, "Serious games—An overview," Univ. Skovde, Skovde, Sweden, Tech. Rep., 2005.
- [8] M. Lytras, C. Lougos, P. Chozos, and A. Pouloudi, "Interactive television and e-learning convergence: Examining the potential of T-learning," in *Proc. Eur. Conf. eLearning*, 2002, pp. 1–11.
- [9] R. Pavlov and D. Paneva, "Interactive TV-based learning, models and standards," in *Proc. HUBUSKA Open Workshop Semantic Web Knowl. Technol.*, 2006, pp. 70–99.
- [10] M. Rey-López, A. Fernández-Vilas, and R. Díaz-Redondo, *A Model for Personalized Learning Through IDTV* (Lecture Notes in Computer Science), vol. 4018. New York, NY, USA: Springer-Verlag, 2006, pp. 457–461.
- [11] F. Belloti, S. Vrochidis, E. Parissi, P. Lhoas, D. Mathevon, M. Pellegrino, et al., "A T-learning courses development and presentation framework," *IEEE Multidisciplinary Eng. Educ. Mag.*, vol. 3, no. 3, pp. 69–76, Sep. 2008.
- [12] M. Zajc and A. Isteni, *Interactive Multimedia T-Learning Environments: Potential of DVB-T for Learning*. Athens, Greece: WSEAS Press, 2009, pp. 103–123.
- [13] S. Walldén and A. Soronen, "Edutainment. From television and computers to digital television," Univ. Tampere Hypermedia Lab., Tampere, Finland, Tech. Rep., 2004.
- [14] T. Sky. (2005). *Tata Sky DTH—Active Learning* [Online]. Available: <http://www.tatasky.com/active-learning-demo-low.html>
- [15] J. Santos, E. Ratamero, J. Arruda, M. Dantas, M. Sanchez, and D. Muchaluat Saade, "RummiTV: An interactive game for the brazilian digital TV system," in *Proc. 26th Symp. Brasileiro Telecomun.*, 2008, pp. 1–6.
- [16] T. Tavares, C. Saibel, C. Bittencourt, G. Mariniello, and C. Santana, "A TV digital interativa como ferramenta de apoio à educação infantil," *Revista Brasileira Inf. Educ.*, vol. 15, no. 2, 2005.
- [17] R. Díaz-Redondo and A. Fernández-Vilas. (2008). *Proyecto SUMA. Estudio Bibliográfico de T-Learning* [Online]. Available FTP: <ftp://ftp.heanet.ie/disk1/sourceforge/s/project/su/sumaproject/Documents/SP3/PT3.5/SUMATarea3.5.1.B.pdf>
- [18] I. Abadía, "Revisión de lineamientos para el desarrollo de contenido educativo para televisión digital interactiva," *Revista S&T*, vol. 10, no. 20, pp. 71–104, 2012.
- [19] BBC. (2012, Nov. 24). *Global Experience Language*, London, U.K. [Online]. Available: http://static.bbc.co.uk/gel/0.2.0/downloads/GEL_web_styleguide.pdf
- [20] (2012, Nov. 24). *OpenCaster 3.1: The Free Digital TV Software* [Online]. Available: <http://www.avalpa.com/the-key-values/15-free-software/33-opencaster>
- [21] A. Prata. (2012, Nov. 24). *iTV Guidelines* [Online]. Available: <http://encyclopedia.jrank.org/articles/pages/6650/iTV-Guidelines.html>
- [22] (2012, Nov. 24). *Xlet Emulator* [Online]. Available: <http://www.xletview.org>
- [23] F. Capra, *El Punto Crucial*. Buenos Aires, Argentina: Editorial Troquel, 1992.
- [24] *Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.2.2*, ETSI TS Standard 102 727 V1.1.1 (2010-01), 2010.

Iván Abadía Quintero was a Graphical Designer from Universidad del Valle, Cali, Colombia, in 2009. His interests in interaction and graphics. He is a member of the i2t Research Group, Universidad Icesi, Cali, and is a Colciencias Young Researcher.





Madelayne Morales Rodríguez was a Telematics Engineer from Universidad Icesi, Cali, Colombia, in 2012. Her interests in project management and planning. She is a member of the i2t Research Group, Universidad Icesi, Cali, in interactive television and DTV areas.



Patricia Madriñán Rodríguez was an Industrial Designer from Universidad Pontificia Bolivariana, Medellín, Colombia, in 1994. She received Graduate in multimedia production from Universitat Politècnica de Valencia, Valencia, Spain, in 1998 and the Ph.D. degree in education with pedagogic mediation emphasis from Universidad LaSalle Costa Rica, San Jose, Costa Rica. She is an Associate Researcher with the i2t Research Group, Universidad Icesi, Cali, Colombia, in the usability and interfaces design areas.



Camilo Ortegón Barajas is a Telematics Engineering student at Universidad Icesi, Cali, Colombia. He is a member of the research seedbed of i2t Research Group.



Andrés Navarro Cadavid (M'95–SM'11) received the Electronic Engineer and master's degrees in technology management from Universidad Pontificia Bolivariana, Medellín, Colombia, and the Ph.D. degree in telecommunications from Universitat Politècnica de Valencia, Valencia, Spain, in 1993, 1999, and 2003, respectively. He is an Advisor of the National Innovation Program on Electronics, Telecommunications and Informatics from Colombian Research and Development System. He has been the Director of the i2t Research Group, Universidad Icesi, Cali, Colombia, since 1999.



Juan Vicente Pradilla Cerón is a Telematics and Informatics Engineer from Universidad Icesi, Cali, Colombia, and received the master's degree in computer science and telecommunications management from Universidad Icesi. He is currently pursuing the Ph.D. degree with Universitat Politècnica de Valencia, Valencia, Spain.