

# Kroster - Juego para Televisión Digital en MHP. Proceso de Desarrollo y Consideraciones de Diseño y de Programación Frente a Aspectos Técnicos

I. Abadía Quintero, M. Morales Rodríguez, *Student Member, IEEE*, C. Ortegón Barajas, J. Pradilla Cerón, P. Madriñán and A. Navarro Cadavid, *Senior Member, IEEE*.

**Title—** Kroster – game for digital television in MHP technology. Process development, design and programming considerations versus technical aspects.

**Abstract—**This article present the development of Kroster, a serious game created for digital television using MHP technology. In the paper we make a description of the process for the game creation, both from the engineering and the design point of view, discussing programming and graphic aspects. We describe some requirements for the development a t-learning game and the restrictions associated with the TV platform, specially the MHP development framework.

**Index Terms—** Digital Television; game; t-learning; MHP; design and programming; technical aspects.

## I. INTRODUCCIÓN

LA televisión digital permite el acceso a regiones apartadas, donde hay mayor biodiversidad y al mismo tiempo se patenta más, la falta de apropiación y conciencia de la misma [1]. Los contenidos de la televisión terrestre tradicionalmente se han ocupado de la “entretenimiento” y muy poco de la educación, entre otras cosas porque educar sin interactuar es una tarea sino imposible, al menos muy difícil.

El despliegue de la TV digital en Colombia, utilizando el estándar europeo DVB-T y posteriormente el DVB-T2 en el año 2009 abre una serie de oportunidades para llegar con contenido educativo interactivo a zonas apartadas de la geografía. El uso del modelo *t-learning* bajo el concepto de juegos serios puede ser una oportunidad interesante en un país con una geografía extensa, donde la televisión y la radio son los únicos medios de comunicación disponibles en muchas regiones.

La posibilidad de desarrollo de aplicaciones interactivas que permite el estándar europeo DVB mediante MHP (*Multimedia Home Platform*) [2], así como las experiencias existentes en el desarrollo de aplicaciones de *t-learning*, bien sea usando MHP, Jinga o herramientas propietarias, ha motivado el desarrollo del trabajo que se presenta en este artículo, que se basa en el desarrollo de un juego educativo interactivo para TV digital empleando MHP y el paradigma de juegos serios. El juego se basa en conceptos de educación ambiental y conservación de especies en peligro de extinción propias del país.

I. Abadía, M. Morales, C. Ortegón, J. Pradilla, P. Madriñán and A. Navarro Cadavid. Laboratorio de Investigación en Informática y Telecomunicaciones i2t. Universidad Icesi. Calle 18 No. 122-135 Pance, Cali – Colombia (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)

La creación de un juego para televisión digital se centra en dos focos de trabajo, programación y diseño; cada área posee una relación intrínseca con los aspectos técnicos del medio televisivo y su aplicación con aspectos de *jugabilidad*. Ambas áreas deben tener como objetivo principal realizar un acercamiento al medio, especificando y determinando los alcances y los obstáculos que se presentan para cada una, con base en la documentación existente y en un proceso de prueba/error (en la mayoría de aspectos). El proceso para el desarrollo de Kroster, en general, fue progresivo, desarrollando partes del juego y readaptando aspectos, tanto de diseño como de programación, teniendo en cuenta los requerimientos que iban surgiendo. Ambos procesos se desarrollaron de forma independiente y se retroalimentaron con base en los resultados que arrojaron las fases de prueba y simulación de cada parte.

Un esquema del proceso de diseño del juego, en general, se puede observar en la Figura 1. El artículo está dividido de la siguiente manera: En la sección II se muestra la motivación y antecedentes del proyecto; en la sección III se hace una descripción del juego y de las consideraciones de diseño tanto en los aspectos técnicos como en los aspectos gráficos; en la sección IV se describe el proceso de programación y las restricciones asociadas a MHP y al entorno de televisión en el desarrollo de un juego; en la sección V se discuten los aspectos de diseño gráfico del juego y las decisiones que se tomaron a lo largo del proceso de desarrollo, con el fin de lograr un aspecto gráfico adecuado y una *jugabilidad* aceptable. Finalmente, en la sección VI se muestran las conclusiones y trabajo futuro.

## II. MOTIVACIÓN Y ANTECEDENTES

Los juegos serios han crecido en importancia en los últimos años. Sus antecedentes se remontan al libro de Clark Abdt, *Serious Games* [3], y han retomado fuerza en los últimos 5 años, gracias al desarrollo de las tecnologías de juegos 3D y las tarjetas gráficas.

En el caso de las aplicaciones para Televisión Interactiva, las tecnologías disponibles, ya sea MHP ó Jinga, así como la capacidad computacional de las cajas decodificadoras (Set Top Boxes - STB), ponen restricciones importantes sobre el desarrollo y uso de los juegos serios en este medio. En los siguientes apartados se hace una breve discusión acerca de los juegos serios y los antecedentes encontrados sobre juegos para televisión interactiva, que motivaron el trabajo que aquí se presenta.

Programación	Evaluación y definición librería TvGame	Desarrollo de clases para el manejo de capas (profundidad)	Construcción de clases para la carga de layers y de imágenes (RoadLayers)	Prueba de clases desarrolladas	Desarrollo e integración primer nivel+mapa+bonus Pruebas y simulación (Xiter/STB)	Construcción de clases para el manejo de slides	Redefinición de clases para el manejo de capas (perspectiva)	Redefinición de clase contenedora de layers en 3 dimensiones	Desarrollo de clases para el manejo de sprites	Desarrollo e integración segundo nivel Pruebas y simulación (Xiter/STB)	Aplicación y adaptación de clases desarrolladas para el tercer nivel	
	Diseño	Bocetación y desarrollo de ideas	Definición de sistema gráfico (tipografía, iconos e identidad del juego) y de parámetros de interacción	Prototipo primer nivel (cálido), bonus y sección de inicio (mapa)		Consideraciones de exportación de imágenes vs peso y correcciones de color y efectos	Construcción de slides para animación	Corrección de tamaño, proporción, definición de imágenes y niveles de profundidad	Desarrollo de menús contextuales		Prototipo segundo nivel (páramo)	Corrección gráfica primer nivel+mapa+bonus
Desarrollo e integración ventanas de ayuda+tercer nivel Pruebas y simulación (Xiter/STB)												

Figura 1. Proceso de creación del juego Kroster

A. Acerca de los Juegos Serios

Un juego serio, es aquel cuyo propósito principal (sin importar su naturaleza) se acerca a aspectos relacionados con la información, la formación y la enseñanza, utilizando el entretenimiento como principal herramienta de comunicación [4]. En general, se pueden encontrar opiniones divididas frente a si los juegos serios se deben encontrar estrechamente relacionados con objetivos educativos [5], [6], o si deben estar relacionados con propósitos informativos y persuasivos enfocados a la formación [7]. Esta clase de aplicaciones poseen características de interacción con el usuario, en las que a través de la diversión, el entretenimiento y en general, el juego y la lúdica, promueven la transferencia de alguna clase de información [8]. Desde el punto de vista de su clasificación, pueden existir juegos de diferentes tipos e índoles, desde los más simples (como juegos de parejas o *trivias*), hasta juegos de alta complejidad que pueden simular una realidad (juegos de simulación, realidad aumentada e inmersión del jugador) [9]. Así mismo, se listan dependiendo del área donde trabajen, ya sea directamente en ámbitos educativos, aunque se pueden encontrar aplicaciones en áreas de salud, militar, transporte y medios informativos [7].

B. En Relación con la Televisión

Aunque no existe, hasta donde conocen los autores, una relación directa entre juegos serios y televisión, sí se han definido dos áreas que promueven el entretenimiento educativo a partir de la interacción con este medio. Por una parte, se encuentra el *t-learning*, como una convergencia de tecnologías que promueven el aprendizaje por medio de la televisión [10]. Esta área en constante crecimiento, asegura que en el aprendizaje a través de la televisión, la interacción surge como un mecanismo fundamental para adquirir y desarrollar conocimientos, favoreciendo capacidades como la comunicación, el análisis y el descubrimiento [11].

Por otro lado, se encuentra el *edutainment* (sigla resultante de la combinación de *education* y *entertainment*), como un concepto que se refiere a segmentos de televisión que se complementan con elementos de aprendizaje [12], lo que convierte la experiencia de aprendizaje en algo más divertido a través de retos y actividades basadas en el entretenimiento [13], [14].

En general, en el entretenimiento educativo a través de la televisión, el usuario/jugador, construye sus propios conocimientos a partir de su contacto con el conocimiento, utilizando varios enfoques: aprender haciendo, aprender de los errores, aprendizaje mediante contacto o relación, juegos de rol y aprendizaje constructivista. Los contenidos,

aplicaciones y juegos pueden tener metas, reglas y modos de competencia distintas y pueden generar posibilidades de placer y oportunidades diferentes que motivan al usuario/jugador a jugar y a entender o asimilar cierta clase de información específica [15].

C. Algunos Antecedentes de Juegos para Televisión

A continuación se hace una breve revisión de algunas aplicaciones educativas para televisión, que ilustran los conceptos expuestos anteriormente:

1. CBeebies.

Propuesto por la BBC de Londres, es un canal de televisión cuyo público son niños menores de 6 años y cuyo principal objetivo es acompañar y reforzar las destrezas preescolares de los niños. En general, el canal transmite contenidos audiovisuales y programas de distintos tópicos, y adicionalmente se puede encontrar contenido interactivo, desde juegos y *trivias*, hasta karaokes, que apoyan la información que se encuentra transmitiendo [16]. La versión web del canal puede ser consultada en <http://www.bbc.co.uk/cbeebies/>.

2. A Turma da Árvore.

Programa de alta audiencia de la televisión brasilera, dirigido a niños de hasta 6 años con problemas de analfabetismo. Fue uno de los pioneros en la televisión sudamericana en incluir aspectos de interactividad a una transmisión de televisión. El programa abarca temas de educación ambiental, salud y conceptos de ciudadanía, por medio de la historia de un grupo de titeres que viven en lo alto de un árbol. La versión interactiva del programa, plantea diferentes canales de comunicación entre el alumno y el profesor, y presenta juegos interactivos que ayudan al desarrollo de la historia, y puntos de quiebre a decisión del usuario [17].



Figura 2. Aplicación *Build a truck*, que invita al niño a construir un camión por medio del control remoto [16]



Figura 3. Juego “Nem todo lixo é lixo”, del programa A turma da Árvore, en el que los niños deben clasificar las basuras en distintos grupos reciclables [17]

3. Rummikub.

Juego matemático proveniente del canal RummiTV, y cuyo principal objetivo es estimular el aprendizaje de operaciones matemáticas y capacidades de razonamiento lógico en niños y jóvenes por encima de los 8 años. El juego consiste en una serie de fichas puestas en un tablero, en donde el jugador debe competir con otros jugadores para deshacerse de todas las fichas que posee. La forma de deshacerse de ellas, es formando grupos de fichas y realizando una serie de operaciones con ellas (por medio del color o de los dígitos que representan) [18].

4. Active Learning.

Es un módulo interactivo desarrollado por la empresa indú DTH-Tata sky, y se encuentra dirigido a niños de entre 7-12 años. Hace parte de una serie de aplicaciones interactivas para televisión, que van desde aprendizaje de inglés, aprendizaje de cocina, juegos y música. Este módulo, es presentado a modo de trivias en diferentes temas, como matemáticas, ciencias, alimentación, entre otras, en donde el niño puede responder, y acumular puntos con los cuales puede reclamar premios [19].



Figura 4. Interfaz gráfica del juego Rummikub [18].



Figura 5. Interfaz gráfica Active Learning. [19].

D. Motivación para Kroster

Los países de otras latitudes por fuera de la zona tropical, centran su visualización del mundo en un ciclo de 4 estaciones, mientras que un país ecuatorial como Colombia, basa su diferenciación climática en una cuestión de altitud. Todo el año se encuentran todos los climas gracias a unas diferencias de altura sobre el nivel del mar. Para alguien acostumbrado a las 4 estaciones puede parecer muy raro un país que las tenga siempre, ya que es un asunto espacial y no temporal. Los animales y las plantas dependen de un hábitat y es así como funciona en este país de contrastes físicos atemporales.

Este fenómeno de los pisos térmicos y la diversidad de fauna que habita a diferentes alturas, desde el nivel del mar hasta las nieves perpetuas por encima de los cinco mil metros de altura, motivó el desarrollo de un juego que permita al jugador aprender de una manera agradable acerca de la fauna existente en los diferentes pisos térmicos.

Así nace la propuesta de “Kroster”, un paseo de “kros” (cross) a través de pisos “tér”micos, en 3 niveles que permiten al usuario, televidente y jugador, tres en uno, hacer el recorrido cambiando de lugar: en primera persona, en tercera persona y con vista cenital o elevada.

E. Consideraciones de Diseño

Kroster es un juego cuya temática fundamental es la concientización hacia la conservación de la biodiversidad. Su objetivo principal de enseñanza es la asociación lógica entre pisos térmicos, sus condiciones climáticas y los ecosistemas presentes en él (fauna y flora). Para lograr este objetivo se usan capas de información e imágenes esquemáticas, que comunican al jugador los conceptos propios de cada piso térmico.

Pero en este proceso de concientización, el cambio de puntos de vista es esencial. Para quien ha nacido en un lugar y nunca ha salido de éste, lo único verdadero es su mundo, no tiene como o con qué compararse y eso hace que no vea las cosas que lo hacen único y dificulta su propia valoración.

**Kroster** es un juego con respiración, de estructuración orgánica: primero el recorrido es subiendo, luego bajando y por último es un remanso, una analogía del ciclo de la vida.

**El inicio:** es en tercera persona, a nivel. Subida por una montaña, desde una perspectiva un poco alejada, pero permite que el jugador vaya tomando conciencia de lo que pasa, viéndolo y causándolo al mismo tiempo.

**El climax:** del juego se logra al llegar a la cima, donde se cambia de posición y debe lanzarse en primera persona por un río.

En la primera parte del juego el movimiento bastante restringido, es muy vertical, hay obstáculos que deben saltarse, usando flecha arriba en el control, mientras el descenso es horizontal, los obstáculos están entre la derecha y la izquierda, y así deben esquivarse por medio de las flechas de un lado a otro.

**Desenlace del juego:** por último se llega al mar, donde la vista cambia a cenital (Figura 6e), ahora el jugador tiene un panorama más completo, lo ve todo desde arriba, combinando lo vertical con lo horizontal, en un juego que va enriqueciendo la experiencia de aprendizaje.

F. Descripción del Juego

Kroster presenta tres niveles, consecuentes con los tres puntos de vista de parte del jugador, que se deben completar

en un tiempo determinado. Al jugador se le asigna un puntaje dependiendo de los elementos recolectados, el tiempo empleado y el logro del objetivo: rescatar a una especie en vía de extinción.

En el primero, el jugador toma el rol de un *bicicrossista* (Figura 6b), en tercera persona, que recorre el piso térmico cálido; en él, debe esquivar obstáculos (i.e., huecos, llantas y manchas de aceite) y recolectar a una serie de frutos autóctonos, propios de este ecosistema; al finalizar, debe rescatar un mono araña. En este nivel, se desarrolla además un bono (Figura 6c) en el que el jugador debe esquivar troncos que van cayendo de un árbol y recolectar una serie de monos, para ganar puntos extra.

El segundo nivel (Figura 6d), presenta al jugador haciendo rafting en el piso térmico de páramo (por encima de los 3000 metros); toma como punto de vista la parte trasera del personaje; el jugador debe esquivar troncos que bajan por el río y recolectar frailejones, que es un arbusto propio del páramo, para al final rescatar a un tapir, que es una especie endémica en peligro de extinción.

Por último, el tercer nivel (Figura 6e), se presenta mediante una vista cenital, el jugador es el capitán de un bote y debe rescatar el mayor número de ballenas jorobadas posibles, en competencia con navíos pesqueros que intentarán pescarlas.

Se puede acceder a cada uno de los niveles, por medio de un mapa principal. En él, se brinda información relevante para cada piso térmico y se resume la puntuación de cada nivel (Figura 6f).

### III. PROCESO DE PROGRAMACIÓN. CONSIDERACIONES FRENTE A ASPECTOS TÉCNICOS Y RESTRICCIONES DEL ENTORNO DE DESARROLLO

MHP [2] es el sistema intermediario (middleware) que fue desarrollado por DVB fórum y estandarizado por ETSI para el desarrollo y ejecución de aplicaciones interactivas en el sistema de televisión digital europeo, que buscaba disponer de una plataforma común que garantizara la interoperabilidad de aplicaciones independientemente del fabricante del Receptor, ya fuera un Set Top Box (STB) o un televisor.

Dada la baja capacidad computacional de los receptores, los desarrolladores de MHP impusieron restricciones importantes en el manejo de gráficos y en el manejo de memoria, por lo que el desarrollo de juegos sobre esta plataforma difiere de forma considerable del desarrollo

sobre otro tipo de plataformas como las consolas de juegos o computadores.

En la práctica, el despliegue de MHP y de dispositivos con capacidad de ejecutar aplicaciones MHP ha sido bastante limitado, lo que impone restricciones adicionales a un desarrollo como Kroster. A esto se suma la poca disponibilidad de herramientas de desarrollo amigables, que permitan que personas con bajos conocimientos de programación desarrollen aplicaciones en MHP.

Con el fin de superar algunas de las limitaciones del estandar MHP, el grupo de trabajo desarrolló la librería *TVGame*, orientada al desarrollo de juegos, que busca simplificar el proceso. *TVGame* es una librería diseñada para desarrollar juegos en *Multimedia Home Platform* [MHP], cuya estructura ha sido construida por el equipo de desarrollo a lo largo de la implementación de Kroster. Esta librería está conformada por una serie de clases que le permiten al desarrollador enfrentar la creación de un juego de forma amigable, contemplando soluciones a situaciones referentes al ciclo del juego y el manejo de capas y animaciones (*sprites* que son elementos gráficos usados en los videojuegos, que permiten un uso más eficiente de los recursos de procesador.). Es una base para el desarrollo de aplicaciones en MHP y está fundamentada en el ciclo básico de un juego (Figura 7), en el ciclo de una aplicación y en los métodos necesarios que se deben definir para que suceda dicho ciclo (Figura 8).

#### A. Aspectos de Programación

La implementación de la aplicación Kroster se desarrolló por medio de la librería *TVGame*, que facilita el manejo de imágenes, capas, colisiones y animaciones en el desarrollo del juego.

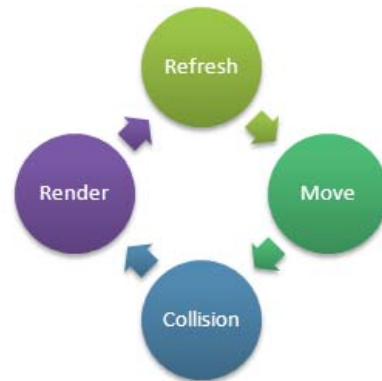


Figura 7. Ciclo básico del juego



Figura 6. a) Identidad del juego - logotipo b) Piso térmico cálido - Nivel 1; c) Bonus nivel 1; d) Piso térmico Páramo - Nivel 2; e) Nivel final. Carrera contra los piratas; y f) Mapa inicial desde donde se acceden a todos los niveles

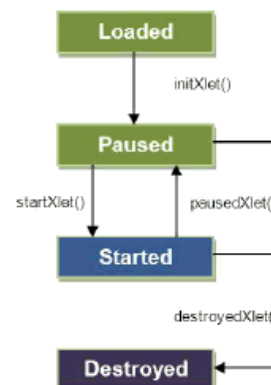


Figura 8. Ciclo de la aplicación

Como se mencionó la idea principal del desarrollo de *TVGame* fue realizar una librería independiente que permitiera su implementación en el desarrollo de juegos en MHP y se consideró la exportación de la misma en un archivo *.jar* para su posterior distribución e implementación. Kroster no fue la primera aplicación que se desarrolló con la librería, pero sí el primer juego completo y con él se identificó la necesidad de agregar y realizar ciertas modificaciones a dicha librería para hacerla más completa y utilizable, conservando siempre su sentido genérico, contemplado desde el principio.

*TVGame* consta de una estructura ligera y genérica, en sus inicios solamente se encontraban las clases básicas para el manejo y disposición de imágenes, que son catalogadas como capas, estáticas y en movimiento en el canvas (i.e *Layer.java*, *ImageLayer.java*, *RoadLayer.java*, *Sprite.java*, *TiledLayer.java*), la carga de recursos gráficos de la aplicación (*Resource.java*) y la clase principal del juego en donde encontramos el ciclo del juego (Figura 7) (*Game.java*).

En Kroster las clases *Layer.java* y *LayerManager.java* están dispuestas para el manejo de capas en cada nivel (e.g., el fondo, el personaje principal y los objetos que interactúan directa o indirectamente con él), dándole libertad al programador para enfocarse en las acciones y los movimientos dentro del escenario, en lugar de centrar su atención en la sincronización de los diferentes elementos y su ubicación en pantalla, como debe hacerse en el MHP estándar. Pero, debido a que el punto de vista del jugador y la interacción del personaje con los elementos del fondo y con los objetos en el espacio son diferentes dependiendo de qué tan alejados se encuentren de la cámara en el nivel (debido al efecto de perspectiva), fue necesario el desarrollo de una clase llamada *PerspectiveUtil.java*, agregada a la librería *TVGame*, que permite el manejo de un mundo visto en tres dimensiones, tal como se ilustra en la Figura 9.

Resultaba razonable que el programador hiciera uso la clase *PerspectiveUtil.java* para darle un aspecto más realista a los elementos, cumpliendo con la restricción existente de no sobrepasar las quince capas en escena; pero el primer nivel (clima cálido) contaba con casi un centenar de elementos en escena, que complicó la tarea del movimiento de forma sincrónica y la ubicación verdadera en pantalla en relación con el jugador. Por esto, esta tarea fue delegada al manejador de capas, sin fusionar sus lógicas, dándole al programador la capacidad de controlar el juego de forma automática (asignando la tarea del manejo perspectivo al *Layer Manager*) o de forma manual (considerando los ajustes necesarios sobre posición en las capas). Sin embargo, aun con esa adición y modificación a la librería para el manejo de perspectivas y manejo sincrónico de las



Figura 9. Efecto de perspectiva en el juego. Aunque el trampolín se encuentra alineado con la palma, el manejo de capas permite superponerlo dependiendo de la posición del jugador

capas no resultó ser suficiente para la implementación del segundo nivel, donde se hace uso de un efecto de 3D en primera persona, en la que los objetos se acercan al jugador a medida que va bajando por el río en el recorrido del nivel.

Para realizar el ajuste constante de tamaño de los objetos, teniendo como referencia la proximidad con el personaje, se presentaron dos alternativas: la primera, sobrecargar el procesador del STB modificando el tamaño de las imágenes en tiempo real, para contar así con un arreglo de imágenes de diferentes tamaños para pintarlas dinámicamente dependiendo de su posición en el nivel.; y la segunda, que resultó ser más efectiva por cuestiones de rendimiento y optimización de recursos, fue establecer que se precargarían únicamente cinco tamaños de una misma imagen. La carga de memoria con imágenes de diferentes tamaños debe realizarse con especial cuidado porque se sale del control de la librería y puede detener la aplicación. Para la ejecución de este proceso de carga de imágenes, se utiliza la clase *Resource.java*, usada en la librería *TVGame* para importar las imágenes. Esta clase cuenta con la característica de importar la imagen tan sólo una vez, de tal forma que si en otro momento se le solicita, la imagen a la que se hace referencia es la misma que se había cargado, optimizando el uso de recursos en el STB. Sin embargo si la imagen cargada originalmente se usa para crear una copia con características alteradas (como es el caso del tamaño) la referencia inicial se pierde y la imagen deja de ser la misma. Esto quiere decir que por cada vez que se modifica el tamaño de una imagen por medio del método *resize()* de la clase *Resource.java*, se entrega un objeto imagen completamente nuevo, con espacio propio en memoria. Por lo tanto, es necesario controlar la operación de cambio de tamaño se realice una única vez y que la imagen sea utilizada en cada uno de los objetos que requieran ese tamaño de imagen.

Para la visualización del río con un efecto de perspectiva, fue necesario tener en cuenta que las imágenes tipo carretera aprovechan su característica de ser repetitivas, tanto en su profundidad como en su simetría; de tal forma que no es necesario almacenar toda la imagen, ya que con solo una esquina se puede formar el resto. Su construcción depende que los detalles de la imagen en la parte superior coincidan con los de la parte inferior, y que los tamaños de estos detalles cambien debido a la inclinación de la imagen. Un ejemplo de esta construcción se puede observar en la Figura 10.

La clase *sprite.java*, encargada del manejo y el funcionamiento de los *sprites*, que son capas del juego cuya función especial es la animación de elementos, pudiendo adoptar o cambiar su forma en cualquier momento. Su

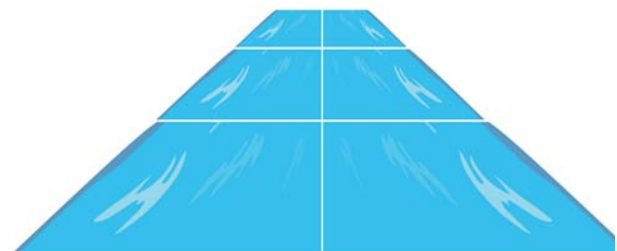


Figura 10. Imagen tipo carretera. Construcción del río en el nivel 2

utilidad se centra en la sencillez de uso, pues al recibir una imagen, son capaces de fraccionarla en pequeñas imágenes como en una línea de tiempo, pintándolas una por una para obtener la sensación de movimiento. Adicionalmente, dichas animaciones se pueden configurar para que se muestren tan sólo algunos de los *frames* y lograr así movimientos definidos para determinadas situaciones, (e.g. saltar, dar una patada, caminar).

Dado que en el segundo nivel los objetos debían salir del horizonte, navegar sobre el río y finalmente sobrepasar el personaje en el flotador, el manejo de capas utilizado en el primer nivel debió replantearse. Debido a que realizar la priorización de las imágenes de forma manual, o a través de una clase organizadora que revisara cuales imágenes tendrían prioridad en la pantalla en un momento específico, resultaba bastante ineficiente (debido a la cantidad de validaciones que debían realizarse para cumplir con las especificaciones); se creó una nueva clase contenedora de layers, con funciones similares a las ya existentes en la clase *Layer Manager*, pero con la posibilidad de manejar tres espacios (frontal, medio y fondo). Se desarrolló entonces la clase de tipo *layer*, llamada *SpacialContainer.java*, que debió ser agregada al *Layer Manager* ya que sobre él recae la responsabilidad de dibujar todas las capas del nivel.

Para lograr los efectos de profundidad y cambio de tamaño de las imágenes de acuerdo con el recorrido, se establece que todas las capas almacenadas en el espacio frontal y en el espacio del fondo nunca cambien su priorización a la hora de pintar (e.g., el personaje flotando y el río); y que los objetos del espacio medio (MIDDLE) no tienen una priorización fija, sino que ella cambia de acuerdo con un nuevo atributo de posición relativa, que puede tomar tres valores: *back*, *middle* y *front*. La priorización de capas se muestra en la Figura 11.

La clase *SpacialContainer*, desarrollada para Kroster, no se incluyó como parte de la librería TVGame, debido a que utiliza objetos que no están concebidos en la librería de juegos, (aquellos con atributo de posición relativa y posición en el eje Z). Además, se implementó una función para el cálculo de las posiciones de los elementos en pantalla (para el cálculo de los ejes X y Y), en función de los siguientes parámetros, ilustrados en la Figura 12.

- Elevación de la visión (A), medida para determinar que tan inclinado se está proyectando el suelo, y calcular así el movimiento que debe realizarse;
- Porcentaje de acercamiento (B), porcentaje de distancia que hay desde el horizonte hasta el personaje (C), y debido a que los objetos sobrepasan la posición del personaje, el porcentaje puede tomar valores negativos;

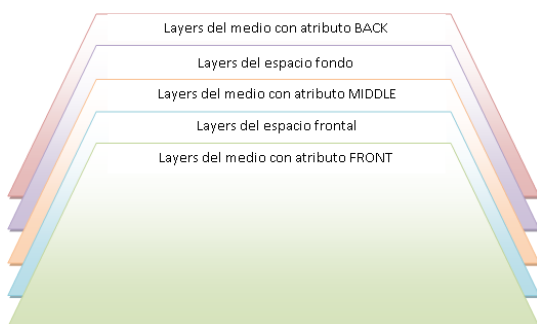


Figura 11. Priorización de capas para el nivel páramo



Figura 12. Parámetros para determinar la posición de un elemento en pantalla

- Objetivo en el eje Y (D), posición en el eje Y donde se encuentra el personaje principal esperando a ser alcanzado por los objetos circundantes;
- Centro sobre el eje X (F), eje que divide la pantalla en dos, izquierda y derecha;
- Punto de origen (G), coordenadas en X y Y de donde parte el objeto; y
- Dimensiones relativas (H), ancho y alto con el que debe pintarse el objeto en un momento específico.

La clase que contiene la anterior función se encuentra implementada en *DimensionalDealer.java*, pero el cálculo del porcentaje de acercamiento se encuentra dentro de la clase *Objeto3D.java*. Ya que estas clases no pertenecen en sí a la librería de juegos TVGame, no deben tener un orden específico por paquetes, ni ser genéricas, sino que pueden ser adaptadas a las necesidades específicas del juego. Las colisiones entre el personaje y los objetos del río se hacen sobre el plano XZ, por lo tanto fue necesario crear también un atributo de profundidad.

A pesar de que la librería de juegos permite redefinir las colisiones como colisiones avanzadas, no fue posible adaptar esta función al segundo nivel, pues para pasar por una colisión redefinida, la librería debe primero revisar las colisiones absolutas sobre el plano XY. Al estar las colisiones en el río en el plano XZ, se generan problemas de manejo entre las dos funcionalidades.

### B. Validación y Pruebas

Dadas las condiciones del sistema MHP y ya que los STB no son computadores de propósito general, el desarrollo de cada uno de los niveles del juego se hace en un computador convencional. Antes de cargar el programa en el STB, este debe ser simulado en un computador que contiene un emulador de MHO, para luego ser probado en un televisor por medio de un Set Top Box (STB) con soporte para MHP.

La simulación se realiza a través de un software de emulación, que le permite al desarrollador tener un ambiente para probar los Xlets (nombre que se le da a un componente de software en MHP, y de forma similar al *applet* en java) que desarrolle y le brinda una idea de su desempeño en un STB real. En Kroster, se utilizó el emulador XleTView [20], cuya configuración y funcionamiento dependen de un archivo principal desarrollado por el equipo de ingeniería del proyecto, el cual no se distribuye junto con el paquete original de XleTView. La simulación en esta herramienta

consiste, básicamente, en hacer uso de la aplicación de manera local para probar su funcionamiento y desplegarla como si se tratara de un STB real. En referencia a Kroster, las pruebas para el desarrollo de la librería se realizaron en el simulador de escritorio, prescindiendo de algunas funciones visuales como rotar una imagen o cargar una imagen traslucida, ya que estas no funcionaban bajo MHP. Dichas funciones (de la clase *Resource.java*) no persisten en la librería y no fueron usadas en el desarrollo.

Para la realización de pruebas en un STB se debe emular el ambiente bajo el cual trabajará la aplicación en una estación transmisora real. Para ello se utilizó el laboratorio de TV del grupo de investigación, que consta de: un transmisor de laboratorio marca Televue, que incorpora el modulador digital con formato DVB, que permite reproducir un flujo MPEG proveniente de una entrada de tipo DVB-ASI/SMPTE -310M a desde un servidor de video por medio de un puerto USB y que convierte el flujo de video en una señal de aire en el rango de frecuencias 55-956MHz y 956-2150 MHz; un STB con soporte MHP, marca Telesystem; dos antenas, una transmisora, conectada al Televue y otra receptora conectada al STB, que capta la señal que es transmitida por la estación base. Adicional a ello, para realizar la transmisión del contenido interactivo, se usa el sistema OpenCaster [21], una colección de herramientas de uso libre diseñadas para ejecutar sobre sistemas operativos Linux, que permite generar, procesar, multiplexar, reproducir y enviar la aplicación interactiva sobre un TransportStream MPEG-2 que se emite al aire, de acuerdo al estándar DVB. Para transmitir la aplicación, es necesario contar con el código fuente de la misma junto con las imágenes y los archivos *.class*, que se generan cuando se construye la aplicación, los que deben ser transmitidos desde el Televue hasta el STB y que se visualizan en la pantalla de salida conectada a la STB para ejecutar el proceso de interacción. El montaje de laboratorio se muestra en la Figura 13.

IV. PROCESO DE DISEÑO. CONSIDERACIONES FRENTE A ASPECTOS TÉCNICOS

La construcción y el desarrollo de una aplicación para televisión digital están condicionados por diferentes factores de tipo técnico, que afectan la forma en que se desarrollan y se conciben creativamente para este medio [22]. En el caso específico del desarrollo de la aplicación Kroster, el primer paso en el ámbito de diseño, fue determinar en papel, las funcionalidades de los elementos, su estructura general (a modo de boceto) y, al mismo tiempo, su jugabilidad (*playability*).

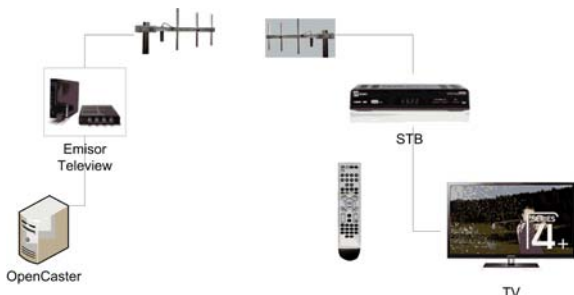


Figura 13. Montaje de laboratorio Kroster

Adicional a ello, se determinó la identidad del juego (logotipo y paleta de colores a utilizar), se definieron tanto sus aspectos iconográficos, como sus características de interacción, y se seleccionó la tipografía a utilizar. La restricción técnica relacionada con el peso de las imágenes para su correcta visualización en el STB, limita en gran medida la construcción de parámetros para la realización de una propuesta gráfica y tiene un peso importante en la definición de aspectos estéticos.

Esto a su vez, restringe la utilización de paletas de colores variadas y de efectos como degradados y sombras. Por esto, las imágenes utilizadas son de orden esquemático, con un alto nivel de contraste, en especial para aquellas cuya legibilidad era importante para la interacción, las cuales fueron delineadas con un borde blanco.

En cuanto a la tipografía, a pesar de que se recomienda el uso de distintas familias tipográficas (e.g., Helvética, Arial o Verdana) para televisión [23], [24], en la práctica, los dos STB de pruebas no permitieron la instalación de fuentes diferentes a las disponibles de fábrica, lo que limitó el uso de diferentes fuentes tipográficas descritas en la literatura y en documentación de MHP. Por tanto, se realizaron los elementos textuales en formato de imagen (Figura 14), garantizando la claridad de la información y la consistencia gráfica del juego. Para elementos explicativos de mayor extensión (como zonas informativas), dada la falta de información referente a cómo otras fuentes tipográficas pueden ser usadas en aplicaciones en MHP, se utilizó la fuente Tiresias, por ser la única disponible instalada en los STB de pruebas.

El proceso de definición de la interacción del usuario, estuvo limitada por los medios de entrada con los que se pretendía que el usuario interactuara, en este caso, el control remoto. El usuario puede manipular el movimiento del personaje en cada uno de los niveles con las 4 flechas de dirección; por otra parte, dos de las teclas de colores y la tecla *enter*, se utilizan para acceder a las opciones de *jugabilidad* (e.g., acceder al menú o a la ayuda, salir del juego, elegir un nivel determinado). En la Figura 15 se puede apreciar la definición de los parámetros de interacción en los principales lugares de navegación del juego: a) mapa principal, para elegir un nivel utilizando las flechas y para entrar al nivel, la tecla *enter*; b) mapa principal, para obtener ayuda la tecla azul y para salir del juego la tecla roja; c) dentro de un nivel, para obtener ayuda la tecla azul y para desplegar el menú, la tecla verde; d) en el menú, para desplazarse entre opciones, flechas arriba y abajo y para escoger una opción, la tecla *enter*; y e) en las ventanas contextuales de ayuda e instrucciones, para cerrar la ventana, la tecla roja y para desplazarse entre *slides*, las teclas derecha e izquierda.



Figura 14. Tipografías utilizadas en elementos informativos dentro del juego

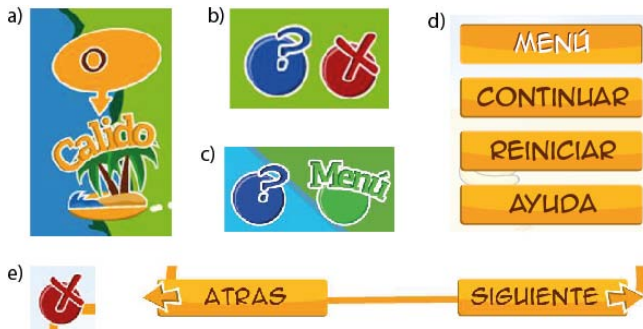


Figura 15. Parámetros de interacción del juego

Al realizar el prototipo del primer nivel, se establecieron los principales elementos en dos niveles de profundidad, para emular espacialidad y la sensación de recorrido: la construcción de un fondo mucho más largo que el tamaño total de la visualización en pantalla que se desplazará lentamente; y la determinación de varios niveles de perspectiva, con algunos elementos estáticos y otros dinámicos (e.g., nubes en movimiento), para lograr la percepción de profundidad.

La mayoría de imágenes se exportaron en formato *PNG* (*Portable Network Graphics*), por su condición de transparencia); otras, como el fondo y la carretera (de tipo mosaico), en formato *JPEG*, para garantizar un menor tamaño de archivo. Para obtener imágenes con un menor tamaño de archivo, dado que se estaba diseñando con paletas de colores reducidas, se optó inicialmente por exportar las imágenes optimizadas para web, con las opciones *PNG-8* con transparencia, reduciendo al máximo la cantidad de colores utilizados (entre 8 y 6). Tras la integración del primer nivel y la simulación local en *XLeTView*, se presentaron falencias en algunos aspectos como la definición de los bordes y problemas en el tamaño y la proporción, como resultado de la exportación (Figura 16). Por esta razón, se modificó de *PNG-8* a *PNG-24*, que aunque aumenta su tamaño de archivo en algunos kilobytes, garantiza una correcta visualización.

Por otro lado, dado que la proporción en la que se trabajó estuvo definida por la resolución de pantalla común en computadores (1024x768), se presentó un inconveniente con la visualización de los elementos en la pantalla del televisor (ver Figura 17); por esto se realizó una corrección aproximada de la proporción y de la ubicación de los elementos en pantalla, basándose en una proporción cercana a los 800x600px. Esta decisión estuvo basada las siguientes razones:

- *XLeTView* no cuenta con la posibilidad de emular en diferentes proporciones de pantalla (por defecto emula en una ventana estándar aprox. de 800x600);
- no existe una forma para emular el cambio de proporción entre una pantalla convencional de computador (4:3) y una pantalla de televisor (16:9);
- no existen herramientas gráficas ni utilidades de



Figura 16. Problemas de definición de las imágenes al optimizarlas en PNG-8



Figura 17. Diferencias de proporción y posicionamiento de los elementos entre la maqueta estática realizada (a) y el montaje en la emulación en *XLeTView* (b)

conversión que permitan exportar imágenes directamente para televisión y que realicen la conversión respectiva de un pixel de computador (1x1) a un pixel de televisión (equivalente a 1,067x1 pixel de computador).

Aunque el primer nivel tenía varios niveles de profundidad, el usuario no alcanzaba a diferenciar los elementos que hacían parte del juego de los simplemente decorativos; por esto, determinó colocar en los elementos relevantes, es decir, los que el usuario recoge durante el juego, un halo luminoso con una animación de movimiento circular (ver Figura 18).

Por otro lado, para animar objetos (e.g., el ciclista) se utilizan *sprites* que por su sucesión, simulan la progresión del movimiento al cambiar de imagen a imagen. Esta sucesión de imágenes se realiza con un número mínimo de imágenes, para garantizar un tamaño pequeño de archivos, pero con la cantidad exacta que permita que se vea un movimiento fluido. En muchos casos, en especial en los niveles 2 y 3, se presentaron inconvenientes para determinar el número de *sprites* requeridos; además, fue necesario agregar un *sprite* en blanco al principio de cada sucesión, para visualizar de forma correcta la animación.

El proceso de diseño se complementó con la realización de los prototipos del segundo y tercer nivel, con los parámetros definidos en la realización y posterior puesta en marcha del nivel uno.

En la integración de cada nivel, se realizaron correcciones y adaptaciones, en especial de cambio de imágenes y detalles gráficos, cuya necesidad se observó durante la simulación y algunas pruebas con usuarios piloto.

Finalmente, se encontró que el agregar funcionalidades adicionales al proceso de navegación (e.g., entrar por medio de una tecla de color y no mediante la tecla *enter* u obtener

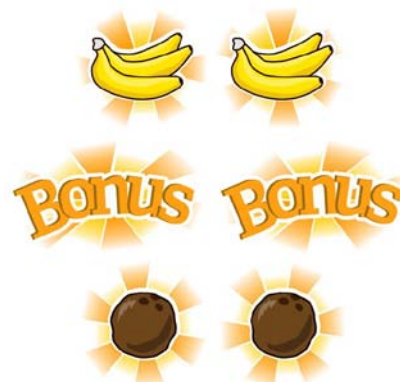


Figura 18. Sprites animados para elementos relevantes en el juego



información mediante una tecla de color y no de forma automática), resultaba ser perjudicial para el uso amigable del juego por parte del usuario, debido a una carga excesiva de información adicional. Por esta razón, fue necesario realizar cambios considerables en las funcionalidades relacionadas con la forma de acceder a un nivel y obtener información acerca del mismo.

#### V. CONCLUSIONES Y TRABAJO FUTURO

El uso de MHP como herramienta para el desarrollo de juegos ha representado un reto en la implementación de clases para el manejo de aspectos de lógica y vistas del juego. En algunos casos fue necesario recurrir a técnicas empleadas en los primeros juegos de computador, para poder cumplir las restricciones de MHP.

Establecer un manejo para las perspectivas que se manejan en la lógica de los juegos es algo que se debe tener en cuenta, y que a pesar que existen vistas clásicas, resulta indispensable que el programador determine esto para facilitar el desarrollo y el despliegue de los gráficos de la aplicación.

En cuanto al diseño, cabe mencionar que si bien existen recomendaciones generalizadas en cuanto a aspectos estéticos, gráficos y de interacción, para la construcción de aplicaciones, no existen lineamientos de orden técnico relacionados con estos aspectos, que permitan a un equipo creativo establecer de manera clara las reglas de creación. Esto a su vez, ocasiona que los procesos de diseño se presenten como procesos iterativos de creación y corrección de aspectos gráficos. A pesar que la realización de este juego, ha posibilitado la aclaración de muchos de esos aspectos que se relacionan con lineamientos técnicos, desde profesiones creativas es indispensable contar con estudios especializados que permitan establecer unas recomendaciones claras en referencia a dichas temáticas, y que sean consecuentes con las nuevas tecnologías y realizando una exploración enfocada hacia los nuevos lenguajes de construcción que se están planteando para televisión (e.g. HbbTV/HTML5 y CSSTV).

En general, podemos afirmar que si bien debe existir una optimización de recursos tanto gráficos como de código, es necesario complementar dicho proceso con el uso de herramientas externas (como OpenCaster) que permitan solucionar problemas relacionados con la implementación de código y de insumos, y con la correcta transmisión de la aplicación.

El proceso de desarrollo de juegos educativos para la TV Digital Interactiva es un trabajo multidisciplinar que debe combinar habilidades técnicas de programación, habilidades de diseño y usabilidad, así como los conceptos pedagógicos asociados al medio empleado.

Como parte de la evolución de este trabajo, en la actualidad se desarrolla una versión de Kroster para HbbTV, que permitirá comparar aspectos técnicos y restricciones de usabilidad e interface entre las dos plataformas de TV interactiva.

#### REFERENCIAS

- [1] F. Capra, *El Punto Crucial*. Buenos Aires, Argentina: Editorial Troquel S. A., 1992.
- [2] ETSI, *ETSI TS 102 727 V1.1.1 (2010-01) - Digital Video Broadcasting (DVB);Multimedia Home Platform (MHP) Specification I.2.2*. Francia: European Telecommunications Standards Institute, 2010.
- [3] A. Clark, *Serious Games*. Lanham: University Press Of America, 2002.
- [4] T. Susi *et al*, "Serious Games - An Overview". University of Skovde, Skovde, Sweden, Tech. Rep. HS- IKI -TR-07-001, 2005.
- [5] A. Derryberry. (2007). *Serious games: online games for learning*. [Online] Disponible: [http://www.adobe.com/resources/elearning/pdfs/serious\\_games\\_wp.pdf](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," *Situated Play: Proceedings of the 2007 Digital Games Research Association Conference*, Tokio: The University of Tokyo, 2007, pp 559-566.
- [7] D. Michael and S. Chen, *Serious games. Games that educate, train and inform*. Mason, OH: Course technology, 2006.
- [8] M. Ulicsak, (2010). *Games in Education: Serious Games*. [Online] Disponible: [http://media.futurelab.org.uk/resources/documents/lit\\_reviews/Serious-Games\\_Review.pdf](http://media.futurelab.org.uk/resources/documents/lit_reviews/Serious-Games_Review.pdf)
- [9] C. Aldrich, *The complete guide to simulations and serious games*. San Francisco, CA: Pfeiffer, 2009.
- [10] M. Lytras *et al*. "Interactive Television and e-Learning Convergence: Examining the Potential of t-Learning," *Proceedings of the European Conference on eLearning*, 2002.
- [11] R. Pavlov and D. Paneva, "Interactive TV-based learning, models and standards," *HUBUSKA Open Workshop Semantic Web and Knowledge Technologies*, Varna, Bulgaria, 2006, págs. 70–99
- [12] S. Walldén and A. Soronen, *Edutainment. From Television and Computers to Digital Television*. Tampere, Finlandia: University of Tampere Hypermedia Laboratory, 2004.
- [13] M. Rey-López *et al*, "A Model for Personalized Learning Through IDTV," *Lecture Notes in Computer Science*, vol. 4018, pp.457-461, 2006.
- [14] F. Bellotti *et al*, "A T-learning Courses Development and Presentation Framework," *IEEE Multidisciplinary Engineering Education Magazine*, vol.3, No.3, pp 69-76, 2008.
- [15] M. Zajc and A. Istenič, "Interactive Multimedia t-Learning Environments: Potential of DVB-T for Learning," *University & industry knowledge transfer and innovation*, págs. 103-123, 2009.
- [16] R. Díaz-Redondo and A. Fernández-Vilas. (2008). *Proyecto SUMA. Estudio bibliográfico de t-learning*. [Online] Disponible: <ftp://ftp.heanet.ie/disk1/sourceforge/s/project/su/sumaproject/Documents/SP3/PT3.5/SUMATarea3.5.1.B.pdf>
- [17] T. Tavares *et al*, "A TV digital interactiva como ferramenta de apoio à educação infantil," *Revista Brasileira de Informática na Educação*, vol.15, No.2, 2005.
- [18] J. Santos *et al*, "RummiTV: An Interactive Game for the Brazilian Digital TV System," *Proceedings of XXVI SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES - SBrT*. Rio de Janeiro, Brasil, 2008.
- [19] Tata Sky. (2005). *Tata Sky DTH – Active Learning*. [Online] Disponible: <http://www.tatasky.com/active-learning-demo-low.html>
- [20] XleTVView. (2011). *Xlet emulator*. [Online] Disponible: <http://www.xletview.org>
- [21] OpenCaster. (2012). *OpenCaster 3.1: the free digital tv softwar*. [Online] Disponible: <http://www.avalpa.com/the-key-values/15-free-software/33-opencaster>
- [22] 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
- [23] BBC. (2012). *Global Experience Language*. [Online] Disponible: [http://static.bbc.co.uk/gel/0.2.0/downloads/GEL\\_web\\_styleguide.pdf](http://static.bbc.co.uk/gel/0.2.0/downloads/GEL_web_styleguide.pdf)
- [24] A. Prata. (2010) *iTV Guidelines*. [Online] Disponible: <http://encyclopedia.jrank.org/articles/pages/6650/iTV-Guidelines.html>



de interfaces gráficas de usuario.

**Iván Abadía.** Diseñador gráfico de la Universidad del Valle (2009) con áreas de interés en diseño de la interacción, interfaces gráficas, multimedia, usabilidad y televisión digital. Hace parte del grupo i2t de la Facultad de Ingeniería de la Universidad Icesi, como Joven Investigador de Colciencias, en áreas relacionadas con la televisión digital en Colombia, sus paradigmas, sus requerimientos técnicos y las posibilidades de este medio dentro de la sociedad colombiana y el desarrollo



usabilidad.

**Patricia Madriñán Rodríguez.** Diseñadora Industrial de la Universidad Pontificia Bolivariana de Medellín (1994), Especialización en Producción Multimedia Universidad Politécnica de Valencia (1998), estudiante de doctorado en Educación con énfasis en mediación pedagógica, Universidad LaSalle Costa Rica. Actualmente es investigadora en el grupo i2t de la Universidad Icesi, en áreas de diseño de interfaces y



investigadores del Proyecto SUCCESS TV [Servicio universal en cooperación Colombia-España para sistemas satélite de televisión] financiado por Colciencias, ejecutado por i2t.

**Madelayne Morales Rodríguez.** Ingeniera Telemática de la Universidad Icesi (2012) con interés en las áreas de planeación y gerencia de proyectos en tecnologías de la información y las comunicaciones (TIC) y servicios interactivos. Desde su último año de carrera es miembro del Grupo de Investigación en Informática y Telecomunicaciones (i2T) de Icesi, en el que trabaja en la línea de comunicaciones inalámbricas. Actualmente hace parte del equipo de



comunicaciones cuánticas.

**Juan Vicente Pradilla Cerón.** Ingeniero de Sistema e Ingeniero Telemático de la Universidad Icesi, y Master en Gestión Informática y Telecomunicaciones con énfasis en Telecomunicaciones de la Universidad Icesi. Actualmente adelanta sus estudios de Doctorado. Sus áreas de interés incluyen las redes inalámbricas y las



semillero de investigación del grupo de Investigación i2t de la Universidad Icesi desde 2011.

**Camilo Ortigón Barajas.** Estudiante de Ingeniería Telemática en la Universidad Icesi. Miembro del



grupo de Investigación i2t de la Universidad Icesi desde 1999, miembro del Centro de excelencia en TIC – ARTICA.

**Andrés Navarro Cadavid (M'95-SM'11)** Ingeniero Electrónico (1993) y Magister en Gestión de tecnología (1999) de la Universidad Pontificia Bolivariana en Medellín, Doctor Ingeniero en Telecomunicación de la Universitat Politècnica de Valencia (2003). IEEE Senior Member, Consejero del Programa Nacional de Electrónica, Telecomunicaciones e Informática, del Sistema Nacional de C&T en Colombia. Director del