

RAICES: un juego serio social para revalorizar las culturas originarias

Autor: Luciano Julián Nomdedeu
Facultad de Informática, Universidad Nacional de La Plata
luciano.nomdedeu@gmail.com

Directores: Javier F. Díaz, Laura A. Fava
Laboratorio de Investigación en Nuevas Tecnologías Informáticas - LINTI
Facultad de Informática, Universidad Nacional de La Plata
jdiaz@unlp.edu.ar, lfava@info.unlp.edu.ar

Resumen. En este trabajo se exponen los aspectos más destacados del proceso de creación de Raíces, un juego serio social que sirve como didáctica para la enseñanza, y busca estimular a los niños de hoy, caracterizados por el uso naturalizado de las tecnologías, a adquirir conocimiento a través de las redes sociales. La propuesta de este juego online tiene objetivos pedagógicos escondidos, objetivos que se pretende, posibiliten a los jugadores obtener un conjunto de conocimientos sobre las culturas aborígenes en forma natural y distendida, despertando curiosidad y deseos por aprender. Se intenta generar un estado emocional que propicie procesos de aprendizaje más efectivos que los métodos tradicionales de enseñanza. Las redes sociales y el tiempo que los niños pasan en ellas, pueden ser más fructíferos si además de favorecer a la socialización, a la cooperación y a la diversión, les permite adquirir conocimiento. El juego Raíces se propone colaborar en el proceso de concientización de los niños y adolescentes sobre los valores de las culturas originarias, sus problemáticas históricas y actuales, y los derechos de estos pueblos, buscando fortalecer la concepción de la Argentina como un país pluriétnico y multicultural.

1 Introducción

En la actualidad, los medios de aprendizaje son muy diferentes a los de hace unas décadas. Las nuevas tecnologías han hecho posible la aparición de nuevos medios para enseñar. Entre ellos, uno de los más novedosos es el del uso de videojuegos y las redes sociales.

Si bien la creación de videojuegos educativos existe desde los inicios de los videojuegos, la evolución de los mismos es constante y día a día se encuentra con nuevos desafíos y oportunidades.

Esta tesina gira en torno a la creación de un juego serio social que busca tomar más atractivo para los chicos el aprendizaje sobre los pueblos originarios de Argentina, de forma que a través del juego, recurso al que están habituados y conocen bien, puedan interesarse y aprender más sobre culturas que son bien nuestras.

1.1 Motivación

Las motivaciones para emprender el diseño e implementación de un videojuego serio para redes sociales son numerosas. Por un lado, se conoce la elevada cantidad de horas que los niños y adolescentes pasan conectados a Internet, compartiendo información con amigos y jugando videojuegos. Esta afirmación está basada en una reciente encuesta realizada para el Ministerio de Cultura de la Nación que reveló que el 57% de los niños argentinos forma parte de una red social y utiliza Internet para comunicarse con sus amigos, divertirse y pasar el tiempo [1]. Por otro lado, se han realizado encuestas en establecimientos educativos urbanos, observando una fuerte tendencia por parte de los niños a jugar videojuegos entre dos y tres horas diarias, con una importante presencia de los juegos de Facebook. Estas encuestas además de indagar sobre los hábitos, usos y preferencias respecto de los juegos, contenían algunas preguntas sobre un tema en particular: los pueblos originarios. En su gran mayoría, se observó un desconocimiento sobre su cultura.

En los últimos años, los aborígenes de la Argentina están comenzando a recuperar el lugar y el derecho que les corresponde como pueblos originarios. A partir de la reforma constitucional de 1994 se reconoce la preexistencia de estos pueblos en el país antes de la colonización española, su derecho a la tierra, identidad, educación, lengua y cultura. Sin embargo, muchas problemáticas persisten, aquí y en todo el mundo, por lo que como complemento de la legislación existente, es fundamental el aporte de la educación y la concientización de los niños y adolescentes para valorar y preservar las culturas de estos pueblos.

Los datos obtenidos a partir de las encuestas, la expansión de las tecnologías de la información y la comunicación (TICs), y el uso cotidiano de las redes sociales, motivaron el diseño de este juego serio social para que funcione como objeto de aprendizaje y posibilite a los jugadores obtener un conjunto de conocimientos y competencias en forma natural y distendida, despertando curiosidad y deseos por aprender.

La expansión viral provista por las redes sociales es otro aspecto que motiva su utilización, permitiendo que el conocimiento y los valores que se quieren transmitir se propaguen con mayor facilidad y alcancen la mayor cantidad de gente posible. El acceso de los niños a las computadoras del programa Conectar Igualdad posibilita también, el uso del juego en un entorno escolar dentro de las aulas.

1.2 Objetivos y metodología a emplear

En este trabajo se expondrán los aspectos más destacados del proceso de creación de Raíces, un juego serio social que sirve como didáctica para la enseñanza, y busca estimular a los niños de hoy, caracterizados por el uso naturalizado de las tecnologías, a adquirir conocimiento a través de las redes sociales.

A través del juego, se intenta generar un estado emocional que propicie procesos de aprendizaje más efectivos que los métodos tradicionales de enseñanza. Las redes sociales y el tiempo que los niños pasan en ellas, pueden ser más fructíferos si además de favorecer a la socialización, a la cooperación y a la diversión, les permite adquirir conocimiento.

El videojuego Raíces se propone colaborar en el proceso de concientización de los niños y adolescentes sobre los valores de las culturas originarias, sus problemáticas históricas y actuales, y los derechos de estos pueblos, buscando fortalecer la concepción de la Argentina como un país pluriétnico y multicultural.

Fomentar el aprendizaje sobre la diversidad cultural, ayuda a evitar la discriminación hacia los habitantes de estos pueblos. Desde el punto de vista de los niños aborígenes, el verse reflejados en un videojuego puede favorecer a que se autoreconozcan como tales sin temor a ser discriminados.

Este desarrollo permitirá no solo poner a prueba y verificar la eficacia de un juego social como objeto de aprendizaje, sino también el potencial del uso de tecnologías estándares Web para el desarrollo de juegos complejos embebidos en redes sociales.

En este trabajo se analizan los criterios que guiaron el diseño del juego Raíces, los distintos frameworks que posibilitaron su creación, los aspectos técnicos de implementación y la integración de los contenidos educativos. Las distintas etapas del proceso de creación y técnicas aquí descriptas, podrán servir para orientar el trabajo de otros grupos que emprendan el desarrollo de juegos serios, y especialmente el de aquellos que creen juegos sociales utilizando tecnologías basadas en HTML5.

2 Juegos Serios

2.1 Definiciones

Si bien la idea de utilizar juegos con propósitos serios existe desde hace varias décadas, la misma parece haber adquirido en los últimos años mayor relevancia y aceptación, resultando en un incremento considerable en la producción de este tipo de juegos. Esta oleada, que algunos autores tales como Djaouti, Alvarez, y otros, sitúan temporalmente a partir del año 2002 [2], coincide con la apropiación por parte de la comunidad científica del término "serious games", o su equivalente en español, "juegos serios".

Los orígenes de este término, se remontan al libro "Serious Games", escrito en el año 1970 por Clark Abt [3], un investigador que tenía entre sus metas la utilización de juegos para entrenamiento y educación. Además de establecer algunos ejemplos sobre la utilización de juegos serios en distintos entornos, Abt presenta una definición sobre este tipo de juegos: "Los juegos pueden ser jugados seriamente o casualmente. Nosotros nos interesamos en juegos serios, en el sentido en que estos juegos poseen un propósito educacional explícito y cuidadosamente pensado, y no han sido concebidos para ser jugados principalmente como modo de entretenimiento. Esto no significa que los juegos serios no sean, o no deban ser, entretenidos."

Esta definición dada por Abt contiene los elementos claves de lo que son los juegos serios, y no difiere notablemente de la dada por Michael & Chen en 2005 y ampliamente utilizada en la actualidad: "los juegos serios son aquellos que no tienen como principal objetivo el entretenimiento o la diversión" [4].

2.2 Discusión actual en torno a juegos serios educativos

Además del debate sobre las posibilidades, ventajas y desventajas que supone el uso de videojuegos dentro de las aulas escolares, existe también una creciente discusión sobre las posibilidades que ofrecen los videojuegos más allá de su uso dentro de las aulas.

Entre las voces que se proclaman a favor de las capacidades educadoras de los videojuegos fuera de las aulas, W. Whright sostiene que actualmente este proceso está ocurriendo de manera invisible y online, cuando los niños vuelven a sus casas y juegan, adquiriendo habilidades y aprendiendo sin darse cuenta, sin ser forzados, siguiendo sus intereses.

En este sentido, J.P. Gee [5] analiza las ventajas que ofrecen los videojuegos comerciales, no serios, en relación con distintos tipos de aprendizaje. Destaca por ejemplo, que juegos como Pokémon pueden motivar a los niños a aprender a leer de manera mucho más eficaz que otros recursos que se les ofrece a la misma edad. En este mismo sentido se orienta la investigación de la pedagoga y docente G. Esnaola [6], quien reconoce ventajas similares en el aprendizaje a partir del uso de videojuegos.

Es importante mencionar, que las capacidades educativas que analizan estos autores no son consecuencia, en la mayoría de los casos, de una decisión por parte de los creadores del juego. Estos juegos, por definición, no son serios, sin embargo posibilitan la adquisición de competencias reales y, en muchos casos, transmiten conocimientos sobre la temática abordada por el juego. Los juegos serios educativos, por el contrario, a pesar de haber sido creados con objetivos pedagógicos, muchas veces no resultan divertidos.

Este problema ha sido reconocido por varios autores: M. Zyda [7], por ejemplo, afirma que “la industria del videojuego ya ha sido testigo del fracaso del edutainment, una combinación extraña de software educacional, levemente impregnado de diálogos tiernos e interfaces similares a las de los juegos”. Según Zyda, a pesar de que la pedagogía es un componente implícito de los juegos serios, la misma debe ser secundaria al entretenimiento, al punto de afirmar que un juego serio que no es divertido no tiene ninguna utilidad, independientemente de su contenido o valor pedagógico. De manera similar, L. Rieber [8] sostiene que: “Un juego serio efectivo, debe tener en cuenta ingredientes motivacionales intrínsecos, tales como desafío, curiosidad, fantasía y control.”

La creación de un videojuego serio que resulte entretenido y atrape a los jugadores es una tarea compleja. Dependiendo del ámbito en que se vaya a utilizar, esta cuestión será más, o menos importante. En ciertos ámbitos, como puede ser la capacitación profesional, la rehabilitación, o aún las aulas escolares, los requisitos de calidad no deberían ser tan estrictos. Con que los juegos ofrezcan cierta ventaja sobre las maneras tradicionales de trabajo en esos entornos, los mismos ya tendrán su utilidad. En estos casos pueden darse dos posibilidades: o los propios jugadores reconocen las ventajas de utilizar el juego para adquirir un beneficio, o el uso del juego ha sido impuesto.

En el caso de los juegos serios “no impuestos” o utilizados fuera de entornos restringidos (i.e. juegos pensados para ser jugados en momentos de ocio), la cuestión es diferente. Estos juegos deberán competir con otras fuentes de entretenimiento. En estos casos de uso “no impuesto”, los juegos deben resultar interesantes y estar al nivel (en cuanto a entretenimiento y calidad) que ofrecen otras alternativas no serias.

En particular, para los videojuegos que buscan transmitir un mensaje y alcanzar cierta masividad, el hecho de que sean divertidos y logren atrapar a los jugadores resulta fundamental. El videojuego Raíces, amén de las posibilidades de su uso en las aulas, fue concebido principalmente como un juego de uso general, no impuesto. Los jugadores de Raíces deberían estar motivados para jugar desde sus casas, preferir este juego antes que a otro (al menos en algún momento), tener deseos de seguir jugando. Por estas razones, el tema de la diversión y su atractivo más allá de lo educativo fue siempre una cuestión central.

3 Diseño del juego

Para emprender la realización de un videojuego, es necesario considerar dos cuestiones fundamentales: sobre qué tratará el juego, y qué tipo de juego será. La primera cuestión hace referencia a la temática del juego, es decir, en qué entorno estará ambientado, quiénes serán sus protagonistas, qué historia se quiere contar. La segunda cuestión, se relaciona con el género del juego y las mecánicas que el mismo presentará. Al hablar de mecánicas, se hace referencia al conjunto de acciones o formas de juego permitidas por las reglas, que guían el comportamiento del jugador, creando interacciones [9].

En algunos desarrollos (principalmente no serios), la elección del tipo de juego surgirá primero y la temática después. En el caso de los juegos educativos y culturales, ocurrirá comúnmente lo inverso. Primero se tiene la temática que se pretende transmitir, y sobre esta temática se escoge el género de juego y las mecánicas que mejor se adapten a la transmisión del contenido. En el caso de Raíces, este último fue el esquema adoptado.

3.1 Búsqueda de un género de juego

Con el fin de identificar los gustos y preferencias de niños de entre 9 y 12 años con respecto al uso de juegos, se elaboró una encuesta que se realizó a más de 300 niños de escuelas públicas y privadas de la región.

El análisis de las mismas, ha permitido una mejor comprensión sobre la relación actual de los niños con respecto al juego. Dentro de este análisis, se observó una fuerte presencia de juegos con elementos del género de plataformas, género en el cual el jugador debe moverse de izquierda a derecha saltando sobre plataformas suspendidas en el aire, mientras supera distintos obstáculos. Con esta característica como base, la mayoría de los juegos identificados presentaba además características distintivas.

En este sentido, una investigación realizada por Nygen et.al. [10], reconoce que dentro de los juegos de plataforma existen diferentes aspectos que pueden resultar

más o menos interesantes para los jugadores. Nygen reconoce tres patrones de preferencia o subgéneros: *Combat* (el jugador debe combatir enemigos), *Flow* (el jugador debe moverse hábilmente por el terreno) y *Puzzle* (el jugador debe explorar distintos caminos y razonar).

Con el fin de que Raíces contemple las preferencias de juego individuales y resulte atractivo para todos, se optó por hacer un juego de plataformas dividido en tres caminos: Combate, Correr y Pensar. Cada uno de estos caminos, además de tener como base los juegos de plataformas, contará con elementos y mecánicas específicas correspondientes a los subgéneros Combat, Flow y Puzzle, respectivamente.

Por otra parte, teniendo en cuenta que el juego sería embebido en la red social Facebook, se decidió incluir características casuales como parte del diseño de Raíces: se aprende a jugar en menos de un minuto, de fácil comprensión, compuesto por partidas breves, gratuito, re-jugable y de carga e inicio accesible y rápido.

3.2 Framework MDA

Cuando se diseña un juego, se piensa inevitablemente en los jugadores, en cómo jugarán, cómo interactuarán, como se divertirán, cómo consumirán ese juego. El framework MDA (Mechanics / Dynamics / Aesthetics) propuesto por Hunicke et.al. [11], es una herramienta que ayuda a comprender las partes involucradas en el diseño de un juego. Este framework divide el diseño en tres aspectos:

- Mecánicas: describe los componentes principales del juego a nivel de representación de datos y algoritmos. Este aspecto se asocia a las “reglas del juego”.
- Dinámicas: describe los comportamientos que ocurren al momento de jugar, basados en las mecánicas y dependientes de las acciones efectuadas por el usuario y su respuesta a lo largo del tiempo. Este aspecto se refiere al “sistema de interacciones” que se crea al momento de jugar un juego.
- Estéticas: Describe las emociones que se espera evocar en el jugador cuando el mismo interactúa con el juego. Este último aspecto se asocia al concepto de “diversión”.

El diseño de la experiencia de juego utilizando el enfoque formal MDA, consiste en un proceso iterativo que tiene su base en reconocer a los juegos como sistemas dinámicos. Estas tres capas de abstracción permiten conceptualizar la forma en que se interactúa con los juegos y a partir de este análisis comprender qué cambios realizar para obtener los resultados deseados. Estos cambios, siempre serán cambios en las mecánicas del juego, que es el único aspecto que puede ser directamente modificado y controlado. Las mecánicas (o reglas), al ser jugadas generan dinámicas, y estas dinámicas son las que a su vez hacen que las emociones emerjan.

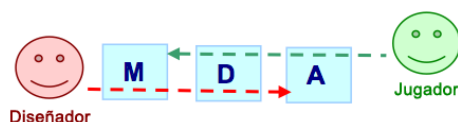


Fig. 1. Direccionalidad del diseño MDA

Raíces fue desarrollado siguiendo este enfoque formal: las mecánicas se fueron refinando iterativamente luego de cada prueba de prototipos, a partir de la observación de las dinámicas generadas, y las emociones que estas dinámicas facilitaban.

Las distintas estéticas descritas en el modelo MDA fueron tenidas en cuenta durante todo el proceso de desarrollo, buscando como fin último generar el mayor rango de emociones posible, potenciando el concepto de diversión y constituyendo un objeto de aprendizaje más efectivo.

4 Motores de juego

Definidas algunas cuestiones básicas sobre el diseño del juego, al momento de emprender la implementación de Raíces, una de las cuestiones centrales fue la elección de un motor de juego (o game engine) que sirviera como base para su programación.

En términos generales, un motor de juego es un sistema de software encargado de gestionar y abstraer aspectos técnicos comunes al desarrollo de juegos.

Cabe aclarar que existe una clara distinción entre lo que es el código del motor de juego, y el código del juego en sí, tal como describe Lewis y Jacobson en su propia definición de motor de juego. En [12], establecen que los motores de juego son una “colección de módulos de código de simulación, que no especifican directamente el comportamiento del juego (lógica del juego) ni el entorno del juego (información de niveles)”.

4.1 Elección de un motor de juego para Raíces

Si bien la mayoría de los motores permite crear cualquier tipo de juegos, algunos están mejor preparados y/o presentan facilidades para ciertos géneros de juego. Por esta razón, antes de elegir un motor en particular, es importante tener definido de qué género será el juego.

La oferta de motores de juego que permiten desarrollar un juego de plataformas es amplia. En el sitio [13], se listan más de setenta motores de juego JavaScript. Esta variedad demuestra la actual expansión que tiene el desarrollo de juegos en HTML5.

La mayoría de estos motores, presentan diferencias importantes entre ellos. Algunos proveen un gran nivel de abstracción y permiten crear juegos de manera sencilla, mientras que otros son más complejos, con menor grado de abstracción, pero a la vez más flexibles. Otra diferencia es el tipo de licencia que presentan: los hay con licencias comerciales, gratuitas y de código libre.

Con el fin de examinar distintas alternativas, se analizaron cuatro motores JavaScript/HTML5 cada uno con distintas características y funcionalidades: Construct 2, Impact JS, Playcraft y Turbulenz.

Luego de analizar y probar en profundidad todos estos motores, se obtuvieron las siguientes conclusiones:

- Construct2 e Impact.js resultan los más adecuados para la creación de prototipos y juegos simples, aunque ambos presentan licencias pagas.
- El uso de Turbulenz para la creación de este tipo de juegos no parece ser la mejor opción debido a su complejidad y bajo nivel de abstracción. Sin embargo, su potencial para crear juegos complejos y las características libres de su licencia pueden hacerlo atractivo para distintos desarrollos.
- El motor Playcraft, aunque promisorio debido a la funcionalidad anunciada, por el momento se encuentra incompleto y no ofrece la robustez de otras alternativas, ampliamente testeadas y con mayor cantidad de usuarios.

A modo de síntesis, en la tabla 1 se expresa el grado en que cada motor satisface los criterios analizados.

Tabla 1. Comparación de motores de juego HTML5

	Construct2	Impact JS	Playcraft Engine	Turbulenz Engine
Abstracción	Muy alto	Alto	Intermedio	Bajo
Flexibilidad	Intermedio	Alto	Alto	Muy alto
Integración con herramientas	Muy alto	Alto	Intermedio	Bajo
Portabilidad	Muy alto	Alto	Intermedio	Intermedio
Licencia	Comercial	Comercial	Libre (con restricciones)	Libre

5 Implementación

La implementación de Raíces ha seguido un enfoque iterativo. En una primera etapa, se crearon prototipos simples para empezar a definir características de la mecánica elegida y testear su funcionamiento. La creación de estos prototipos, supuso una implementación básica de un nivel de cada subgénero: Combate, Correr y Pensar. Para esto se utilizó el motor Construct2 y gráficos provisionales.

Estos primeros prototipos fueron puestos a prueba desde un primer momento, permitiendo así identificar de manera temprana las fortalezas y debilidades del juego propuesto. La flexibilidad y facilidad de uso de Construct2 hacen de este software una herramienta ideal para la creación de prototipos, permitiendo el mismo realizar ajustes rápidos en el diseño de niveles y variables de la mecánica. Al escalar la complejidad del juego, sin embargo, el mecanismo de programación en bloques de Construct2, imponía ciertas limitaciones.

Partiendo de este prototipo básico de un nivel por subgénero, se empezó a realizar una implementación algo más desarrollada del juego (con varios niveles por subgénero) utilizando otro motor: Impact.js.

La transición a este motor, requirió una re-implementación completa de las mecánicas y niveles existentes. La forma de programar juegos con Impact.js, al seguir un paradigma orientado a objetos, resulta distinta de la forma de trabajo con Construct2.

Más allá de lo relativo a la implementación, las cuestiones de diseño de juego desarrolladas durante el trabajo con Construct2 constituyeron un gran aporte y sirvieron de base para el desarrollo inicial con Impact.js.

La posibilidad de escribir código, motivó la generación de comportamientos más complejos y reutilizables. En esta etapa, el prototipo fue adquiriendo una forma más cercana a lo que sería el juego final: varios niveles encadenados, de dificultad creciente, y con entidades claramente definidas y funcionales a las necesidades de cada nivel y del juego como un todo. Aquí también comenzó a implementarse una interfaz de usuario básica.

Si bien el avance utilizando este motor fue grande, las restricciones comerciales de su licencia traían aparejada la imposibilidad de distribuir el código fuente del motor. Debido a las características académicas de este desarrollo, el uso de este motor para la implementación definitiva del juego no era la mejor opción.

En la búsqueda de un nuevo motor para realizar una re-implementación de Raíces, una de las opciones más atractivas en ese momento era la del motor de juego Playcraft, pero luego de probarlo, la gran presencia de errores y el pobre rendimiento gráfico al ejecutar el juego en ciertas computadoras, motivaron la búsqueda de una alternativa más robusta y eficiente.

Después de evaluar las características de otras alternativas de código libre disponibles, se decidió probar el motor de juego de código abierto Turbulenz. Aunque relativamente nuevo, contaba con algunos juegos de gran calidad como carta de presentación y ejemplos que demostraban su alto potencial. Las primeras pruebas realizadas con este motor arrojaron buenos resultados en lo referente a rendimiento, funcionalidad y robustez, por lo que se definió su utilización.

En particular, Turbulenz mostró una de sus principales fortalezas en su motor de física, que aun ofreciendo un amplio rango de posibilidades, resulta eficiente y liviano.

En contraposición a esto, el módulo de renderizado no siempre se mostraba del todo eficiente. Al avanzar una re-implementación de un nivel de Raíces utilizando este motor, se observó que el proceso de renderizado demoraba más tiempo que algunas de las otras alternativas probadas previamente. Pensando a futuro, con una complejidad gráfica creciente y una necesidad de compatibilidad con distintas computadoras, esta desventaja se iría profundizando. En adición a esto, Turbulenz no resulta compatible de manera nativa con navegadores sin WebGL, restringiendo aún más la compatibilidad. Esta dependencia de WebGL se producía únicamente debido al motor de renderizado.

Frente a estas cuestiones, se empezó a gestar la idea de reemplazar este aspecto conflictivo de Turbulenz utilizando un motor de renderizado más eficiente e independiente de WebGL. La organización de Turbulenz en módulos poco acoplados hacía viable este reemplazo.

Entre los motores de renderizado HTML5 disponibles en ese momento, Pixi.js destacaba sobre el resto por ciertas características: de código abierto, orientado a la eficiencia y multiplataforma. Pixi.js, en comparación con el motor de renderizado de Turbulenz, ofrecía numerosas ventajas, por lo que se decidió su utilización.

Para realizar el reemplazo del módulo de renderizado, era necesaria la creación de una arquitectura que gestionara la integración de Pixi.js y el resto de los módulos de Turbulenz de manera transparente. Esta arquitectura, debía además, proveer soporte para la creación de entidades de juego y niveles de una manera sencilla.

Para el desarrollo final de Raíces, se decidió, por lo tanto, crear un motor de juego propio, escrito en JavaScript y basado en Pixi.js y Turbulenz, el cual brindara abstracciones de alto nivel y supliera las carencias de este último en relación a los aspectos de renderizado, compatibilidad con distintos navegadores y equipos, portabilidad, e integración con herramientas externas (i.e. editor de niveles, editor de diálogos, generadores de contenido multimedia).

5.1 Módulos de alto nivel

Contar con abstracciones de alto nivel, posibilitan y facilitan el desarrollo del juego en sí mismo. Si bien estos módulos han sido creados especialmente para sustentar el desarrollo de Raíces, los mismos podrían a su vez ser reutilizados en otros desarrollos de juegos y no se limitan únicamente al género de plataformas.

A continuación se hará una descripción de las funcionalidades y características esenciales de cada uno de los módulos que componen el motor de juego desarrollado.

- **Entity Manager:** la clase Entity constituye la estructura base de todos los objetos que participan del juego y tienen una imagen, una animación y un cuerpo físico asociados. Cada entidad tendrá también, en la mayoría de los casos, un comportamiento asociado. Este módulo brinda las bases para permitir la creación de entidades de una manera sencilla, y abstrae comportamientos comunes a todas las entidades de juego.
- **Game Manager:** cumple la importante tarea de asegurar la ejecución del videojuego como un todo. Este administrador funciona como el controlador principal de la ejecución del juego y se comunica con el resto de las partes del motor.
- **Physics Manager:** encapsula el motor de física provisto por Turbulenz.
- **Input Manager:** encapsula los mecanismos de entrada provistos por Turbulenz para el manejo de interacción con teclado y mouse, y los provistos por Pixi.js para el manejo de interacción táctil.
- **Sound Manager:** ofrece una interfaz simplificada para la creación y reproducción de sonidos por parte de las entidades de juego, y para la reproducción de música global.

- **ViewPort:** maneja el estado actual de la cámara del juego (zoom y posición) y el tamaño de la pantalla.
- **Animation2D:** provee abstracciones para la creación de animaciones y sus variables de control asociadas (i.e. velocidad, repetición).
- **Collision Map:** define la parte sólida de cada nivel, es decir, los espacios en los cuales el jugador podrá o no desplazarse. Posee métodos auxiliares tanto para la creación como para la identificación de las zonas de colisión.
- **Level Loader:** su principal tarea consiste en leer y procesar archivos de definición de niveles. Posee además un mecanismo de carga anticipada de niveles y caché, para agilizar el proceso de pasaje de un nivel a otro.
- **Level Events Handler:** desde este módulo, puede definirse eventos específicos para cada nivel, que se ejecutarán mediante invocación explícita, o al cumplirse cierta condición.
- **Dialog System:** este subsistema se encarga de generar a partir de archivos de diálogos externos, un grafo con todas las posibilidades de comunicación existentes entre dos personajes determinados. Se encarga también de controlar el flujo de ejecución de cada diálogo en ejecución.
- **Background:** permite definir fondos multicapa con paralaje, generando una ilusión de profundidad en entornos 2D. Se encarga de controlar automáticamente la repetición horizontal de cada imagen que constituye el fondo.
- **UI Manager:** ofrece las bases para diagramar pantallas de interfaz dinámicas y autoajustables a distintos tamaños de pantalla. Permite definir animaciones de posición, tamaño y opacidad para objetos de interfaz de usuario.

5.2 Integración con herramientas externas

Editor de niveles.

Por lo general, cada uno de los niveles de un juego, suele definirse en un archivo propio que contiene toda la información específica que lo distingue del resto. Esta información puede incluir desde valores de configuración simples, hasta información sobre el terreno y de las entidades a crear, considerando en este último caso, la posición de cada una en el escenario y sus valores de inicialización.

Luego de realizar algunas pruebas entre varios editores de escenarios 2D, se decidió utilizar *Tiled Map Editor* debido a su alta funcionalidad y versatilidad.

El módulo de carga de niveles desarrollado, incluye funcionalidad para leer de estos archivos: información del nivel (nombre, tipo, vidas, etc.), información de entidades (posición, tipo, tamaño, atributos de inicialización simples y complejos), información de fondos, e información de colisiones (de distintos tipos, a partir de figuras geométricas rectangulares o poligonales)

Al guardar el archivo de nivel desde *Tiled Map Editor*, toda esta información se empaqueta y queda disponible para ser leída por el motor de juego. Cuando el módulo de carga de niveles procesa este archivo identificando los distintos componentes, los

datos estáticos de definición de nivel se transforman en entidades activas con las que el usuario puede interactuar.

Editor de diálogos.

El sistema de diálogos diseñado para Raíces, permite que dos personajes entablen una conversación. Esta conversación, no seguirá un único camino lineal, sino que dependerá de las elecciones realizadas por el jugador y del estado actual del juego.

De manera similar al enfoque utilizado para incorporar niveles generados por una herramienta externa, el grafo de diálogos que representa cada conversación, también puede ser construido a través de un editor. En este caso, la herramienta elegida fue *yEd Graph Editor*, un editor visual de grafos de propósito general.

El subsistema de diálogos desarrollado, permite identificar en estos archivos: ramificaciones condicionales, menús de elección múltiple y lanzamiento de eventos, además de distintos atributos para cada nodo (duración, avance automático, etc.).

6 Contenido y evaluación

Desde un primer momento y a lo largo de todo el desarrollo, Raíces se planteó como un videojuego que lograra un equilibrio entre dos cuestiones: por un lado debía ser entretenido y atrapar a los jugadores, y por otro, debía estimular y facilitar el aprendizaje del tema pueblos originarios.

La investigación teórica realizada, contribuyó a reafirmar la postura de que para desarrollar un videojuego educativo que aproveche al máximo las posibilidades del medio, es imprescindible la comunión entre estas dos cuestiones.

Por un lado, un videojuego educativo con poco contenido, no aportará demasiado. Por otro, un videojuego con un alto grado de contenido, pero con falencias en el diseño de la jugabilidad y la integración de los mismos, probablemente no generará deseos de ser jugado, desaprovechando lo más atractivo del medio.

En esta sección se analizará la forma en que se planteó el aspecto educativo del videojuego Raíces y las principales etapas del proceso de elaboración e introducción de este contenido como parte del juego.

También se describirán las distintas pruebas realizadas con los destinatarios de este videojuego, evaluando los resultados obtenidos tanto en los aspectos de jugabilidad, como en los pedagógicos, describiendo los cambios y decisiones derivadas de estas pruebas.

6.1 Formas de introducir contenido educativo

El aspecto "serio" de Raíces está dado por un doble objetivo: el de motivar a los niños para que se acerquen e interesen por la temática pueblos originarios, y el de enseñar contenidos específicos sobre la historia y cultura de estos pueblos.

En primer lugar, Raíces debía ser percibido como un juego, más allá de su temática y su contenido educativo. Una vez captado el interés y la atención de los niños, el contenido se iría desprendiendo de manera natural e incremental.

En este sentido, es deseable que los contenidos a transmitir se integren directamente con las mecánicas del juego y no sean vistos por los jugadores como algo impuesto o forzado. En el caso de Raíces, la introducción de contenido educativo como parte de la mecánica principal del juego, se estructuró en distintos ejes:

- **Visual:** este aspecto ha sido aprovechado principalmente a partir de la definición de personajes y escenarios representativos de cada pueblo.
- **Sonoro:** la música de Raíces ha sido compuesta en relación a la región y cultura del pueblo originario que participa del relato.
- **Narrativo:** El principal recurso narrativo utilizados en Raíces es el diálogo. A lo largo de los niveles, se suceden diálogos y eventos que siguen un hilo histórico.
- **Simbólico:** M. Treanor et.al. [14] analiza el potencial que tienen los videojuegos para generar "sentido" a través de las reglas, y como esto puede ser utilizado para comunicar ideas.

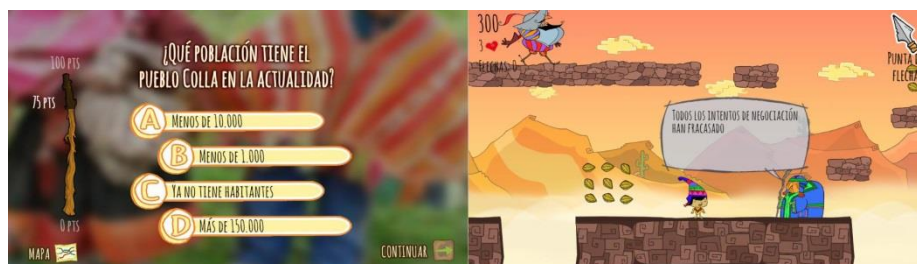


Fig. 2. Capturas del videojuego Raíces

Además de la introducción de contenido dentro de cada nivel del juego, se decidió aprovechar también las pantallas de carga entre niveles. En estas pantallas aparecen contenidos que se relacionan con los temas vistos en los niveles, pero que por una u otra razón, no pudieron ser incorporados de manera directa a los mismos. Cada vez que se completa un nivel, antes de avanzar al siguiente, se muestra una pantalla inter-nivel que podrá tener algunas líneas de información o una pregunta en forma de trivia para evaluar la apropiación de conocimiento por parte de los jugadores.

6.2 Pruebas

Para un primer testeo, se convocó en la Facultad de Informática a niños de entre 9 y 13 años. Los niños pudieron probar durante aproximadamente una hora, un prototi-

po, realizado en Impact.js, con los tres subgéneros implementados: Combate, Correr y Pensar. Se advirtió gran entusiasmo por pasar cada nivel y llegar al final del juego. Finalizado el testeo, los niños pidieron seguir jugando y se entusiasmaron con la idea de poder jugar más niveles en el futuro.

Al finalizar la prueba se les suministró a los niños una pequeña encuesta para determinar las preferencias por los distintos subgéneros: el 33% respondió que el subgénero Pensar fue el que más le gustó, el 45% optó por Combate y el 18% por el subgénero Correr. Si bien hay diferencias entre ellos, cada subgénero tuvo sus adeptos.

Al contar con una versión más elaborada de Raíces, se realizó un segundo testeo, al que asistieron 12 niños y niñas. Además de observarse sus reacciones con el juego en sí, se prestó especial interés a su interacción con los aspectos educativos recientemente añadidos. Con respecto a esta cuestión, pudo observarse que algunos niños prestaban atención a los textos de información, mientras que otros solo hacían una lectura rápida, lo que motivó un cambio en la forma en que se presentan estos textos.

A partir de las preguntas cualitativas de una nueva encuesta, se pudo apreciar un interés de los jugadores por tener más niveles, más personajes y más escenarios. En algunos casos, destacaron también el hecho de que el juego les permitiera aprender sobre las culturas originarias.

7 Conclusiones

Como parte de este trabajo, se analizaron los criterios que motivaron cada aspecto del diseño del juego, los motores de juego HTML5 utilizados, los aspectos técnicos concretos de la implementación de Raíces, y la integración de contenidos educativos como parte misma del juego. El juego en cuestión puede verse en el sitio: www.raiceseljuego.com.ar.

Para cada una de estas etapas, se ha descripto oportunamente un marco teórico, y se han expuesto los desafíos encontrados y su forma de resolución.

A la complejidad de realizar un videojuego serio elaborado, debe sumarse además la complejidad que supone utilizar una tecnología relativamente nueva como HTML5. La principal dificultad encontrada durante el desarrollo de Raíces se presentó con respecto a esta última cuestión: para poder tener en Raíces todas las características que se deseaba tenga el juego desde su concepción inicial, fue necesario la implementación integral de la gran mayoría de ellas, no contenidas en otro motor de juego HTML5. Por un lado, esto permitió la creación de un motor de juego propio, reusable y con una gran funcionalidad, aunque también amplió considerablemente el tiempo de implementación, retrasando la planificación original. Al momento de planificar la creación de un videojuego serio, debe tenerse en cuenta la complejidad de cada una de las etapas y la cantidad de tareas distintas involucradas en la concreción de cada una de ellas, planteando a partir de este análisis un diseño de juego acorde a los plazos y recursos disponibles.

La experiencia del trabajo realizado en Raíces y los paradigmas presentados en este informe, podrán resultar de utilidad para aquellos interesados en el diseño de juegos serios de distintos tipos.

Con respecto al juego resultante y su uso como objeto de aprendizaje, en esta tesis se propusieron distintas formas de introducción de contenido, que fueron puestas a prueba con resultados satisfactorios en diversas instancias de testeo. La integración de estos contenidos como parte del juego y no como algo forzado, fue un factor determinante para la aceptación del juego como un todo por parte de los chicos, que pudieron aprender y tomar conciencia sobre el valor de la cultura de los pueblos originarios, sin dejar de lado la diversión.

En relación a la integración de Raíces con redes sociales, si bien la misma es, por el momento, acotada, el diseño integral del videojuego se realizó teniendo en cuenta este aspecto social e incluyó características en este sentido (niveles cortos, no tiene final, pueden agregarse nuevos niveles continuamente, sistema de puntos competitivo, personalización del avatar, etc.), por lo que está preparado para que se exploten muchas otras funcionalidades provistas por este tipo de redes.

Más allá de los aspectos de Raíces que puedan seguir siendo completados y mejorados, este desarrollo ha permitido verificar principalmente dos cuestiones: desde lo técnico, las capacidades de las tecnologías HTML5 para la creación de juegos complejos embebidos en redes sociales; y desde lo educativo, la potencialidad de los videojuegos serios web como objetos de aprendizaje.

Referencias

1. [Online] Encuesta Nacional de Consumos Culturales
<http://sinca.cultura.gob.ar/sic/publicaciones/libros/EECC.pdf> (último acceso 20/05/2016)
2. Djaouti, D., Alvarez, J., Jessel, J. P., & Rampoux, O. (2011). Origins of serious games. In *Serious games and edutainment applications* (pp. 25-43). Springer London.
3. Abt, C.C. (1970) *Serious Games*. Viking Press, New York. ISBN: 978-0819161482
4. Michael, D. R., & Chen, S. L. (2005). *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade. ISBN: 978-1592006229
5. Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1), 20-20.
6. Esnaola Horacek, G. et al. (2014). Videojuegos en la educación. *Aularia*, 3(1) Enero. pp: 21-26.
7. Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9), 25-32.
8. Rieber, L. (1996). Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational Technology Research & Development*, 44 (2), 43-58.
9. Fullerton, T., Swain, C., & Hoffman, S. (2004). *Game design workshop: Designing, prototyping, & playtesting games*. CRC Press. ISBN: 978-1578202225
10. Nygren, N., Denzinger, J., Stephenson, B., & Aycock, J. (2011, August). User-preference-based automated level generation for platform games. In *Computational Intelligence and Games (CIG)*, 2011 IEEE Conference on (pp. 55-62). IEEE.

11. Hunicke, R., LeBlanc, M., & Zubek, R. (2004, July). MDA: A formal approach to game design and game research. In Proceedings of the AAAI Workshop on Challenges in Game AI (Vol. 4).
12. Lewis, M., & Jacobson, J. (2002). Game engines in scientific research. *Communications of the ACM*, 45(1), 27-31.
13. [Online] HTML5 Game Engines Wiki
<https://github.com/bebraw/jswiki/wiki/Game-Engines> (último acceso 20/05/2016)
14. Treanor, M., Schweizer, B., Bogost, I., & Mateas, M. (2011, June). Proceduralist Readings: How to find meaning in games with graphical logics. In Proceedings of the 6th International Conference on Foundations of Digital Games (pp. 115-122). ACM.