

DESARROLLO DE SOFTWARE CRUCIGRAMAS PARA MOVILES ANDROID

Universidad Nacional José María Arguedas (UNAJMA)

Abstract

Hoy en día, la industria de los videojuegos es uno de los sectores más rentables y estables a nivel mundial, cuyo crecimiento se incrementa a medida que pasan los años. Uno de los factores que tienen permitió al sector para tomar impulso, ha sido la expansión hacia otras plataformas como que son teléfonos inteligentes, debido al crecimiento de su popularidad entre los usuarios.

Este proyecto tiene como objetivo el desarrollo de la lógica de un videojuego de plataformas género en los teléfonos inteligentes que tiene el sistema operativo Android. Esto cubre el manejo del principal carácter, de los enemigos, los escenarios, las colisiones entre los elementos, el ciclo de vida, etc.

Palabras Clave

Plataforma android es un sistema operativo basado en el núcleo Linux, Android Studio es el nuevo IDE (integrated Development Environment – entorno de desarrollo integrado) que Google ha puesto a disposición de los desarrolladores de Android de forma gratuita, AndEngine un framework para desarrollar videojuegos en Android.

Introducción

CRUCIGRAMA es un juego de palabras cruzadas para un jugador.

Consiste en formar y/o completar sobre el tablero palabras que se crucen entre sí, haciendo el uso de frases relacionadas con la palabra y hace poder obtener dicha palabra, donde el plataforma Android, Android studio y AndEngine nos ha servido para poder desarrollarlo, ya que en el ambiente de los videojuegos de hoy en día son tecnologías adaptables y su uso sencilla y también se podría decir que es la más acogida.

El uso de los videojuegos con fines educativos es un campo en auge en los últimos años. Se han realizado estudios sobre sus beneficios en la enseñanza, no sólo por su carácter atractivo y motivador para el estudiante, sino también por la gran variedad de campos y materias que pueden abarcar así como las habilidades que pueden entrenar. Considerando estos beneficios como premisa ¿es posible extraer alguna conclusión interesante del análisis de la gran cantidad de datos que se generan en un juego como resultado de la interacción y de la propia acción de jugar? Es más, en un juego educativo ¿es posible utilizar estos datos para conocer cómo está siendo el aprendizaje e incluso tratar de predecir sus resultados?

Métodos

El juego crucigrama es el juego que tiene un objetivo educativo implícito o explícito para que los niños aprendan algo específico. Un objetivo que explícitamente programa el maestro con un fin educativo, o la persona que lo diseña, ya sea el educador, el maestro, el profesor de apoyo, los padres, los hermanos mayores, los abuelos, los amigos, etc., y está pensado para que un niño o unos niños aprendan algo concreto de forma lúdica.

En la enseñanza formal, en la escuela, es un método de enseñanza, una forma estructurada para instruir o enseñar los contenidos escolares. El juego crucigrama, es un juego que espontáneamente realizan los niños sin un objetivo educativo, pero podemos transformar en educativo en el momento que lo diseñamos para que ejerciten o aprendan contenidos educativos y académicos. Son actividades lúdicas con el objetivo de estimular el desarrollo evolutivo del niño preescolar, en ese sentido son juegos didácticos, pensados para enseñar o estimular aspectos del desarrollo evolutivo.

En el momento que existe un objetivo educativo, un objetivo de aprendizaje deja de ser juego y pasa a ser trabajo o tarea escolar, o aprendizaje. En ese sentido podemos decir que el juego crucigrama es el extremo del juego, del juego espontáneo de los niños, o lo que normalmente, coloquialmente entendemos por jugar, por juego. Ya que jugar por definición no tiene objetivos educativos explícitos, de ahí que aparezca el término juego educativo, cuando se utiliza como material que ayuda a aprender, como un método de enseñanza. Un método que busca situaciones lúdicas para enseñar los objetivos educativos.

Metodología

Programación extrema

La programación extrema es una metodología de desarrollo ligera (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas.

Este modelo de programación se basa en una serie de metodologías de desarrollo de software en la que se da prioridad a los trabajos que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación.

Una de las características principales de este método de programación, es que sus ingredientes son conocidos desde el principio de la informática. Los autores de XP han seleccionado aquellos que han considerado mejores y han profundizado en sus relaciones y en cómo se refuerzan los unos con los otros. El resultado de esta selección ha sido esta metodología única y compacta. Por esto, aunque no está basada en principios nuevos, sí que el resultado es una nueva manera de ver el desarrollo de software.

El objetivo que se perseguía en el momento de crear esta metodología era la búsqueda de un método que hiciera que los desarrollos fueran más sencillos. Aplicando el sentido común.

Java

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general.

Java es un lenguaje de programación orientado a objetos desarrollado por la Sun Microsystems, una compañía famosa sobre todo por sus estaciones UNIX de faja alta. Inspirado al C++, Java fue proyectado con la finalidad de obtener un producto de pequeñas dimensiones, simple y portátil sobre diferentes plataformas y sistemas operativos, sea a nivel

de código fuente que a nivel de código binario; lo que significa que los programas Java pueden ser ejecutados sobre cualquier computadora en la cual sea instalada la máquina virtual.

NetBeans

El NetBeans IDE es un entorno de desarrollo - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

Android

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma.

El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java.

Android studio

Google sigue anunciando novedades en su conferencia I/O 2013. Una de ellas ha sido Android Studio, un IDE especializado para Android que llega para sustituir a Eclipse a la hora de crear aplicaciones.

Android Studio está basado en IntelliJ IDEA, un IDE para Java de JetBrains, así que no empiezan de cero: de hecho, la base es realmente buena. Sobre eso han desarrollado características específicas para desarrollar en Android.

Por ejemplo, Android Studio ofrece la posibilidad de ver en directo los cambios al diseño de las aplicaciones en las diferentes resoluciones que soporta Android. La misma interfaz nos permite ver cómo aparece la aplicación con cada traducción.

También tenemos información útil en el margen de la pantalla de código, como iconos o colores según los mencionemos en el código. Además, Android Studio incluye integración con Google Cloud Messaging, para no tener que salir del IDE.

En general, Android Studio parece una muy buena herramienta. Google por fin toma la iniciativa para lanzar un IDE propio, y la base de IntelliJ IDEA es realmente buena (y superior a Eclipse desde mi punto de vista). Habrá que ver si Google anuncia más novedades que lo hagan aún más interesante.

AndEngine

AndEngine, un framework para desarrollar videojuegos en Android

AndEngine es un Engine 2d usando OpenGL para juegos para la plataforma Android.

- **BaseGameActivity:** El BaseGameActivity es la raíz del juego, que contiene el motor y crea la vista donde se va a dibujar todo. Hay siempre exactamente un solo Engine por cada BaseGameActivity.
- **Engine:** El Engine es el motor interno del juego, se encarga de ir dibujando en pantalla y actualizando objetos en la escena, que contiene todo el contenido que tu juego lleva.

Normalmente hay una escena por Engine, a menos que vayas a usar un SplitScreenEngines.

- IResolutionPolicy: Una implementación de IResolutionPolicy interface es parte del EngineOptions. Te hace abstraerte de la resolución del terminal, tú trabajas para una resolución y el AndEngine se encarga del resto.
- Camera: Un objeto Camera define el rectángulo visible actualmente de la escena actual, no tiene por qué ser la escena completa. Normalmente hay una cámara por escena. Hay subclases específicas que permiten hacer zoom y mover la cámara suavemente.
- Scene: La clase Scene es el contenedor para todos los objetos que se van a dibujar en la escena. Una escena puede tener Layers, que son capas para ordenar objetos. Hay subclases de la Scene como CameraScene/HUD/MenuScene que tienen comportamientos específicos.
- Entity: Una entidad es un objeto que puede ser dibujado, como Imágenes, rectángulos, Texto, Líneas. Una entidad tiene posición/rotación/zoom/color...
- Texture: Una textura es una imagen que se guarda en memoria. En Android, las imágenes deben ser una potencia de 2.
- ITextureSource: Una implementación de ITextureSource-interface se encarga de cargar una imagen en una posición en la textura.
- TextureRegion: Una TextureRegion define un rectángulo en una imagen. Las TextureRegion se usan por Sprites para usar una imagen grande en la que guardamos muchas imágenes pequeñas.
- PhysicsConnector: Motor de físicas integrado en el Engine

Jdk

Java Developers Kit

Para desarrollar programas en Java, el programador debe tener un compilador Java y bibliotecas de compilación, junto con las bibliotecas de apoyo para tareas de programación habituales, tales como la depuración. El Kit de desarrollo de Java representa la colección de estas bibliotecas, incluyendo el "javac" el compilador Java. El JDK incluye el compilador, las bibliotecas de programación y el JRE con el fin de brindar a los programadores una plataforma que se puede compilar y ejecutar programas Java.

Sdk

Un SDK (Software Development Kit), o kit de desarrollo de software, es un conjunto de herramientas que ayudan a la programación de aplicaciones para un entorno tecnológico particular. Es decir, las aplicaciones desarrolladas sobre el SDK estarán destinadas a algún sistema operativo, plataforma hardware, consola de videojuegos o paquete de software en especial.

Son muchos los recursos que puede contener un SDK. Detallemos algunos de ellos:

- Una interfaz de programación de aplicaciones (API). Puede verse como una abstracción del funcionamiento interno del entorno sobre el que vamos a trabajar. Se trata de un conjunto de funciones, rutinas, estructuras de datos, clases y variables que nos permiten manipular el mecanismo de la plataforma sin conocerlo internamente.
- Un entorno de desarrollo integrado (IDE). Un editor que nos ayuda a escribir fácilmente el código fuente del programa. Generalmente, también brinda una interfaz amigable para dos aplicaciones fundamentales:
 - Debugger. Permite "testear" el programa en cada paso de su ejecución.

- **Compilador.** Traduce el código fuente a lenguaje de máquina, obteniendo así un programa ejecutable.
- **Código de ejemplo y otra documentación.** Como punto de partida para empezar a desarrollar aplicaciones.
- **Un emulador del entorno.** Por ejemplo, si desarrollamos una aplicación para móviles desde una computadora de escritorio, nos permite saber cómo la vería el usuario final.

Actualmente, plataformas como los sistemas operativos Android, iOS y Windows Phone ofrecen kits para desarrollar software que funcione sobre sus entornos, y muchas redes sociales tienen SDK específicos para desarrollar todo tipo de aplicaciones en diferentes lenguajes.

Casos de uso

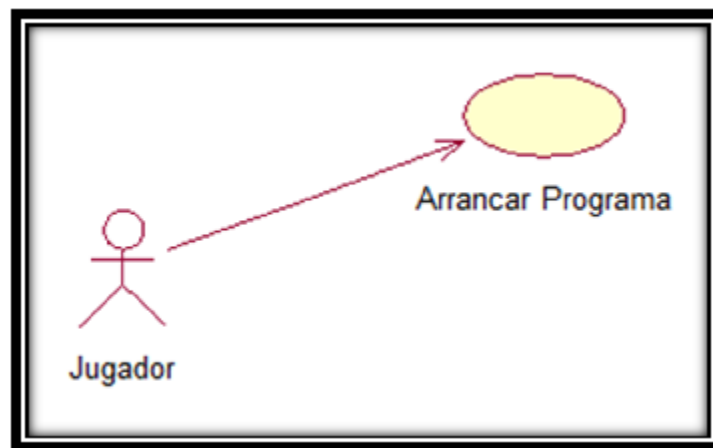


Fig 1

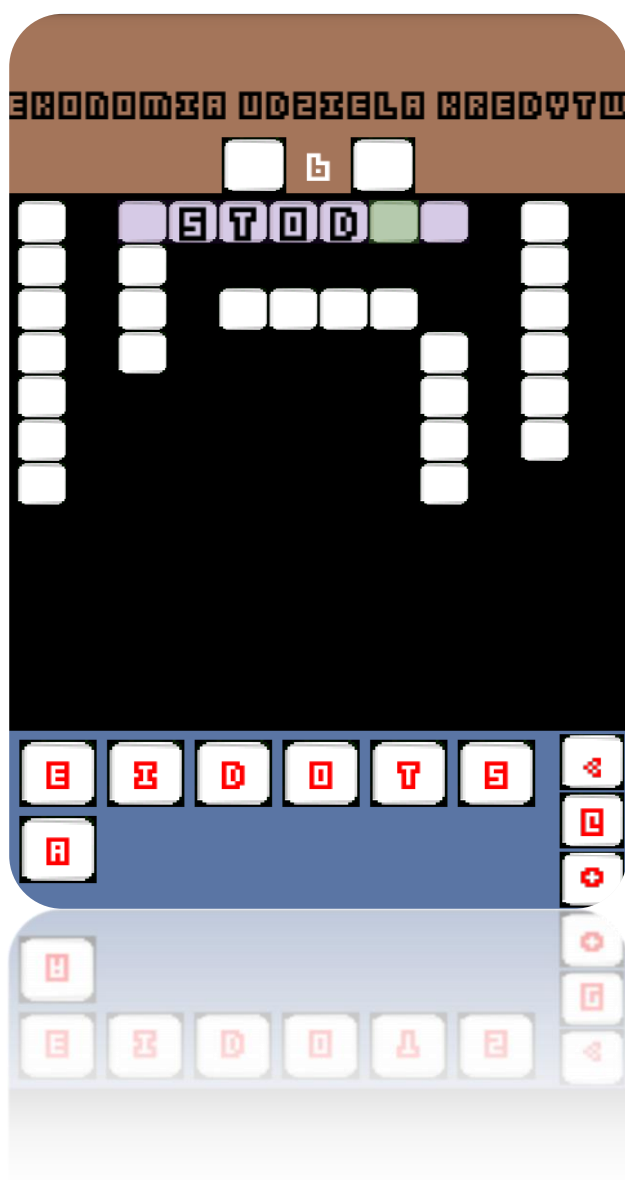
Resultados

Los propósitos trazados que nos planteamos contribuyeron significativamente en el mejoramiento de la calidad de los aprendizajes.

Ha permitido ejercitar a los jugadores en los procesos de desarrollo del pensamiento matemático y comunicativo (concreto, representativo y abstracto).

El compromiso asumido y la dedicación del grupo han generado un cambio de actitud, hacia estas áreas sirviendo como muestra para su aplicación en las demás áreas.

Se ha logrado integrar áreas curriculares y Los jugadores se sintieron motivados.



Conclusiones

El presente trabajo es un aporte muy importante para nosotros, no fue tanto como deseábamos pero concluimos con resultados importantes.

Nosotros intentamos transmitir cómo mediante el juego los alumnos pueden aprender de una manera más práctica y activa en su educación, sin olvidar el gran atractivo del juego impidiendo así que los niños pierdan interés en la materia. El juego va exigiendo una cierta dificultad cada vez mayor, pero ese esfuerzo haciéndolo agradable, aceptado y comprendido por cada uno, ofrece una visión clara de los resultados que se esperan con él.

Pensamos que al incluirse el juego en las actividades diarias de los alumnos se les va enseñando que aprender puede ser fácil y divertido, que se pueden generar cualidades como la creatividad, el deseo y el interés por participar, el respeto por los demás, atender y cumplir reglas, ser valorado por el grupo, actuar con más seguridad y comunicarse mejor, es decir, expresar su pensamiento sin obstáculos. Por todo esto pensamos que el juego nos puede ayudar en nuestro camino como docentes, sacando el máximo partido a lo que nos rodea y ayudando a nuestros alumnos con los juegos que mejor se adecuen a ellos.

Agradecimientos

Queremos mostrar nuestro agradecimiento a todas aquellas personas que nos han ayudado en nuestro largo y duro camino como estuante. Sin ellas, ni el presente manual ni nuestros aprendizajes podrían haber visto la luz.

Referencias

[1] Android studio

<https://developer.android.com/sdk/index.html>

<http://www.elandroidelibre.com/2013/05/asi-funciona-el-nuevo-android-studio-pasos-previos.html>

[2] Algoritmo

<http://blackshell.usebox.net/archive/algoritmo-para-generar-crucigramas.html>

[3] AndEngine

<http://www.aprendiendodeandroidymas.com/2012/05/utilizando-el-framework-andengine.html>

<http://www.genbetadev.com/desarrollo-aplicaciones-moviles/andengine-un-framework-para-desarrollar-videojuegos-en-android>

[4] Repositorio de código fuente de AndEngine

<https://github.com/nicolasgramlich/AndEngine/tree/GLES2-AnchorCenter>

[5] Video tutoriales para la instalación

<https://www.youtube.com/watch?v=CwPCwwydE68>

<https://www.youtube.com/watch?v=uvv23lNOqyY>

Datos de Contacto:

- *Ricardo Huamani Allcca,*
- *Eber J. Apahuasco saccaco y*
- *Diego M. Escalante Coronel*

Universidad Nacional José María Arguedas (UNAJMA)

Ricardohuamani04@gmail.com

Eber582_89@hotmail.com

Coronel_14@gamil.com