Communication Papers
Media Literacy & Gender Studies

CP, 2015, Nº6, pp. 63-73. ISSN 2014-6752. Girona (Catalunya). VÁZQUEZ-CANO, Esteban y FERRER DELGADO, Desiderio: La creación de videojuegos con Scratch en Educación Secundaria. *Recibido: 25/06/2015 - Aceptado: 10/07/2015*

LA CREACIÓN DE VIDEOJUEGOS CON SCRATCH EN EDUCACIÓN SECUNDARIA

Creating videogames with Scratch in Secondary Schools

AUTORES: VÁZQUEZ-CANO, Esteban y FERRER DELGADO, Desiderio

Universidad Nacional Española a Distancia (UNED) – España
Consejería de Educación de Castilla-La Mancha – España

evazquez@edu.uned.es, dferrerde@gmail.com

Resumen

Este artículo presenta una experiencia educativa en la que alumnos de Bachillerato crean videojuegos en el aula mediante software de programación libre. El proyecto educativo tiene su base en las aplicación informática "Scratch", un entorno gráfico de código abierto del Massachusetts Institute of Technology para aprender a programar de forma intuitiva por bloques. Como resultado de la experiencia, los estudiantes han sido capaces de diseñar y crear videojuegos complejos con diferentes módulos de programación aumentando así sus destrezas técnicas y promoviendo una mayor creatividad del proceso de enseñanza-aprendizaje. En un horizonte no muy lejano, el analfabetismo digital vendrá representado por aquellas personas que no son capaces de ser creadores digitales. Una creación digital que constituye una competencia necesaria para el correcto y completo desenvolvimiento de una persona en cualquier ambiente académico, social y profesional y que ayuda a desarrollar objetivos y competencias de la enseñanza formal desde una perspectiva más creativa.

Palabras clave

Videojuegos, programación, Educación, Scratch.

Abstract

This paper presents an educational experience in which high school students create video games in the classroom through free programming software. The educational project is based on the software application "Scratch," an open source graphical environment developed at the Massachusetts Institute of Technology to learn intuitively coding by blocks. As a result of the experience, students have been able to design and create complex video games with different programming modules thus increasing their technical skills and promoting greater creativity of the teaching-learning process. In a near horizon, digital illiteracy will be represented by those who are not able to be digital creators. A digital creation that is a necessary competence for the proper and complete development of a person in any academic, social and professional environment and to develop objectives and competencies of formal education from a creative perspective.

Key words

Video games, programming, Education, Scratch.

1. Introducción

La mayoría de los proyectos de videojuegos en el aula se basan en su uso didáctico principalmente desde principios didácticos inspirados en la gamificación (Gamification) o en el aprendizaje basado en juegos (Game-Based Learning) (Zichermann, & Cunningham, 2011; Deterding, 2012; Sheldon, 2012; Kapp, 2012). Estas iniciativas incorporan al aula el videojuego como un recurso didáctico en apoyo de diferentes asignaturas y contenidos pero recientemente se están incorporando nuevos proyectos basados en el movimiento global liderado principalmente por Code.org y CodeAcademy que promueven la enseñanza de la programación informática en las escuelas, poniendo el foco en las excelentes posibilidades didácticas y laborales que se abren para los jóvenes que sepan programar (Code.org, 2013; Nicks, 2014; NPR Staff, 2014).

La enseñanza de programación en las escuelas no es nada nuevo. Lenguajes de programación

como "Logo" surgieron a finales de los años 60 y se convirtieron en un potente atractivo para esos "jóvenes programadores" que se sentaban por primera vez delante de un ordenador. La programación informática es un proceso que promueve un pensamiento complejo y el desarrollo de diferentes competencias claves al posicionar al estudiante ante procesos de autocorrección y búsqueda de errores (depurar un programa que no funciona adecuadamente), también los enfrenta a retos de resolución de problemas complejos (introduciendo al alumno en la algoritmia) o les presenta conceptos que pueden llegar a ser complejos para un alumno y que requieren del trabajo en equipo y cooperativo para solucionar problemas como la recursividad. Asimismo, es un proceso que si se diseña de forma apropiada incita a la creatividad. Un grupo de estudiantes ante un reto de crear algo nuevo desde su imaginación, mediante el acuerdo y el consenso y la distribución de roles es una actividad altamente significativa para desarrollar contenidos y competencias en el aula con alumnado adolescente.

Si un alumno de ingeniería llega a la universidad con conocimientos básicos de física, ¿por qué no llega también con conocimientos básicos de programación? ¿Tiene sentido que su primera experiencia en el campo de la programación sea, directamente, en la universidad? La respuesta a esta pregunta es algo que podemos encontrar, por ejemplo, en el manifiesto por la educación en ciencias de la computación en el siglo XXI (The Guardian, 2012):

> *"Creemos que todos los niños deberían tener la oportunidad de aprender ciencias de la computación, empezando en la escuela (…) Enseñamos física básica a cada niño, no con el objetivo principal de educar físicos si no porque todos ellos viven en un mundo gobernado por sistemas físicos. De la misma manera, todos los niños deberían aprender un poco de informática desde temprana edad porque van a vivir en un mundo en el que la computación está en todas partes."*

En este artículo presentamos y analizamos una experiencia de creación de videojuegos en el aula de Bachillerato desde una doble perspectiva: didáctica y técnica.

2. La programación en el aula. Una tendencia mundial

Enseñar de manera creativa significa adoptar enfoques imaginativos para hacer el aprendizaje más interesante, emocionante y efectivo (Xinogalos, 2012; Wilson, Hainey, & Connolly, 2013). El uso de los videojuegos ofrecen experiencias que promueven satisfacciones intrínsecas y ofrecen oportunidades para el aprendizaje auténtico (Nelson, 2009; Smith, 2010; Scaffidi, & Chambers, 2012). Existe una tendencia mundial que considera la programación en el aula como una actividad de presente y futuro para el desarrollo de competencias relacionadas con la realidad del mundo laboral y personal de los estudiantes. De esta manera, países como Francia han empezado este curso escolar 2014/15 a programar en la Educación Primaria (LeJDD, 2014), al igual que Reino Unido tanto para la Educación Primaria como para la Educación Secundaria. Precisamente, en Reino Unido se está viviendo una "revolución tecnológica" en las escuelas y entidades privadas están impulsando "Code camps" (campamentos y cursos de programación para adolescentes) (Jones, 2014), o subvencionando la compra de placas Arduino y Raspberry Pi para que los docentes no se vean frenados por la falta de recursos y puedan formar a los escolares en ciencias de la computación. Otros países como Estonia, llevan ya tiempo enseñando programación a los alumnos de primaria (BBC, 2014), Australia lo hará próximamente y países como Finlandia, Israel, Corea del Sur, Nueva Zelanda o Grecia están desarrollando programas piloto.

En Estados Unidos, figuras clave de la industria tecnológica como Bill Gates, Mark Zuckerberg o Jack Dorsey apoyan el proyecto Code.org (http://code.org/) que busca, precisamente, concienciar a alumnos y profesores en las ventajas de la enseñanza de la programación en las escuelas. En España, y especialmente, en Castilla-La Mancha, inspirados por el movimiento global de la Hora del Código (*Hour of Code*, que busca enseñar a programar a partir de los cuatro años). Se considera que al introducir al estudiante tempranamente en las ciencias computacionales, tendrán una base para el

éxito en cualquier carrera del siglo XXI (http://studio.code.org/). La "Hora del Código" es un movimiento global que llega a decenas de millones de estudiantes en más de 180 países. Cualquier persona, en cualquier lugar puede organizar un evento con base en la programación. Existen tutoriales de una hora disponibles en más de 30 idiomas. No se necesita experiencia ni la edad es un problema. La Figura 1 muestra la prolífica actividad de eventos llevados sobre computación y programación desarrollados a nivel mundial por esta iniciativa.

Figura 1. Eventos mundiales sobre programación en la Hora del Código

El uso de la programación en las aulas va más allá de una necesidad social o de proyección profesional de los estudiantes y estudios recientes han demostrado que aprender a programar tiene un impacto positivo en la creatividad y respuesta emocional de niños con dificultades de aprendizaje (Clements, & Swaminathan, 1995; Gold, 2011; Sánchez-Montoya, 2011), así como en el desarrollo de las habilidades cognitivas y socio-emocionales (Burke, & Mattis, 2007; Calder, 2010; Kordaki, 2012; Fessakis, Gouli, & Mavroudi, 2013).

Asimismo, el uso de herramientas digitales y entornos virtuales convierten el proceso de enseñanza-aprendizaje en una actividad más social y colaborativa al promover una mayor interacción para su desarrollo. Investigaciones en el inicio del siglo XXI ya demostraban que cuando los niños se encuentran trabajando con un ordenador hablan usando más del doble de palabras por minuto que cuando se encuentran realizando otras actividades sin utilizar dispositivos electrónicos (Wartella, & Jennings, 2000). Además, cuando los niños están trabajando con el ordenador es más probable que busquen la asistencia y los consejos de otros compañeros, incluso si hay un adulto presente, incrementando la socialización entre los compañeros (UKCRC, 2010; LTS, 2010). Asimismo, cuando se cuenta con un ordenador por estudiante, los niños tienden a formar grupos mientras trabajan con dispositivos tecnológicos (Druin, 1998http://l3d.cs.colorado.edu/~ctg/classes/techcog02/readings/kidsasinformants.pdf). Por último, existen evidencias científicas que demuestran que el alumnado que aprende a programar en edades tempranas tiene menos estereotipos de género en relación a las carreras científicas: Ciencias, Tecnología, Ingeniería y Matemáticas (Burke, & Mattis, 2007); y menos reticencias para continuar sus estudios y profesiones en estas disciplinas (Matlin, 1993; Rae, 1996).

A lo largo de las últimas tres décadas, grupos de investigación y universidades han desarrollado diferentes herramientas que han tratado de facilitar el aprendizaje y la enseñanza de la programación, de manera que el alumnado no tenga que aprender una sintaxis compleja y pueda centrarse en los conceptos informáticos sin tener que preocuparse por errores de compilación. En este sentido, el trabajo desarrollado por el grupo "Lifelong Kindergarten" en el "Media Laboratory" del MIT destaca por encima del resto, desarrollando herramientas como "Scratch" y "App Inventor", que cuentan con cientos de miles de usuarios en todo el mundo. Sin embargo, a pesar de todas las ventajas que parece indicar el uso de la programación como herramienta didáctica, también existen estudios (Hazzan, Gal-Ezer, & Blum, 2008) que ponen de manifiesto que se requiere un profesorado bien formado para conseguir aprendizajes significativos y útiles, evitando que se convierta más bien en un pasatiempo o una moda pasajera. Aprender a escribir código, sin más, es como aprender a deletrear. Para que se consigan los beneficios esperados se requiere trabajar la lógica y profundizar en los conceptos fundamentales de la programación además de conocer sus funciones más sencillas.

3. Scratch: tecnología libre e intuitiva para programar

"Scratch" es un lenguaje de programación que facilita crear historias interactivas, juegos y animaciones y compartir sus creaciones con otras personas en la Web. Es un entorno de programación visual y multimedia basado en "Squeak" destinado a la realización y difusión de secuencias animadas con o sin sonido y al aprendizaje de programación. Este lenguaje de programación se ha desarrollado en el "Lifelong Kindergarten" del Media Laboratory del MIT de la Universidad de California en Los Ángeles (https://scratch.mit.edu/).

Esta aplicación, que forma parte del software de las XO y también se utiliza con otros sistemas operativos, ofrece posibilidades educativas a través de un entorno que hace que la programación sea más atractiva y accesible para todo aquel que se enfrente por primera vez a aprender a programar.

Figure 2. Página de inicio de Scratch

Scratch está basado en el lenguaje de programación LOGO y sus características más importantes son:

- Está basado en bloques gráficos y la interfaz que tiene es muy sencilla e intuitiva.
- Tiene un entorno colaborativo mediante el cual se pueden compartir proyectos, scripts y personajes en la web.
- El trabajo en Scratch se realiza mediante la unión de bloques que pueden ser eventos, movimientos de gráficos y sonidos.
- Los programas pueden ser ejecutados directamente sobre el navegador de Internet.

Sus ventajas son varias:

- Es un programa gratuito, de software libre.
- Es perfecto para enseñar y aprender a programar a niños o adolescentes o a cualquier persona sin conocimientos informáticos de programación.
- Está disponible para varios sistemas operativos, Windows, Mac, Linux y es multiplaforma (se puede utilizar con dispositivos digitales móviles: smartphones y tabletas).
- Permite compartir los proyectos a través de Internet, pudiendo ser descargados y utilizados por otras personas.
- Es multilenguaje.

4. Scratch para fines educativos

El diseño del lenguaje de programación Scratch es un lenguaje visual y no hay que escribir líneas de programación, por tanto se evitan los errores al teclear; se pueden realizar todo tipo de proyectos y actividades personalizadas. Para conseguir que estos objetivos sean posibles, los creadores de Scratch (Resnick et al. 2009) han introducido tres principios o características básicas en el diseño de este lenguaje de programación. Estos principios son: que la lengua de programación sea lúdica, significativa y social.

El lenguaje de programación debe ser lúdico. La idea es que la lengua de programación facilite el juego y que se puedan probar, con facilidad, diferentes opciones. Scratch tiene unos "bloques de programación" de diferentes colores, con conectores que permiten que se puedan encajar unos con otros. El objetivo es que los niños puedan jugar con ellos desde el principio y probar a construir sencillos programas (Resnick et al. 2009; Lamb, & Johnson, 2011).

La experiencia al utilizar el lenguaje de programación debe ser significativa. De este modo, en el diseño de Scratch, sus creadores, han dado prioridad a dos criterios del diseño: *diversidad* (que pueda soportar diferentes tipos de proyectos: historias, juegos, animaciones, simulaciones) y *personalización* (que los proyectos se puedan personalizar importando fotos, voces, gráficos, etc.)

El uso de la lengua de programación debe propiciar la interacción social. El desarrollo de Scratch va muy unido al desarrollo de su página Web. Para que Scratch tenga éxito necesita que una gran comunidad de personas comparta, apoye, critique, colabore y pueda construir sobre el trabajo de otros. Así el concepto de "compartir" está construido en el entorno de interfaz de usuario de Scratch. Haciendo un clic sobre "compartir" los proyectos suben a la página Web de Scratch para ser compartidos. Otras personas apoyarán, criticarán y construirán sobre los proyectos de otros, el objetivo es que finalmente resulte una experiencia de aprendizaje interactiva y enriquecedora para todos (Resnick et al. 2009; Kim, Choi, Han, & So, 2012).

El uso de este programa puede servir para contribuir al desarrollo de algunas habilidades de orden superior y directamente relacionada con la adquisición de competencias clave de los estudiantes. Entre las habilidades que Scratch puede contribuir a desarrollar estarían (Bright, 1991; Scaffidi, & Chambers, 2012):

- *Análisis*: La capacidad para distinguir y separar las partes de un todo hasta llegar a conocer sus principios o elementos.
- *Síntesis*: Capacidad para llegar a la composición de un todo a partir del conocimiento y reunión de sus partes.
- *Conceptualización*: La capacidad de abstraer los rasgos que son necesarios y suficientes para describir una situación, un fenómeno o un problema.
- *Manejo de información*: Capacidad para visualizar y ubicar los datos y la información necesarios para la mejor comprensión de un fenómeno o situación dada; discernir la pertinencia de datos e informaciones disponibles y encontrar tendencias o relaciones entre conjuntos desordenados de datos o informaciones.
- *Pensamiento sistémico*: La capacidad para visualizar como un sistema los elementos constitutivos de una situación o fenómenos, así como la habilidad de visualizar los sistemas como totalidades que forman parte de totalidades mayores y que pueden ser descompuestos en totalidades menores. Operativamente implica las capacidades de análisis y síntesis pero agrega el carácter dinámico y se centra en el estudio de las interacciones.
- *Pensamiento crítico*: Capacidad de pensar por cuenta propia, analizando y evaluando la consistencia de las propias ideas, de lo que se lee, de lo que se escucha, de lo que se observa.
- *Investigación*: La capacidad para plantear interrogantes claros con respecto a una situación o fenómeno dado; de proponer hipótesis precisas y modelos conceptuales de lo que se estudia; de producir o recopilar datos e información con el propósito de verificar el modelo conceptual y las hipótesis; se examina el peso y la validez de la información y el grado con el que se refutan las hipótesis o los modelos conceptuales y, por último, formular teorías, leyes o conceptos acerca del fenómeno en estudio.
- *Metacognición*: La capacidad de reflexionar sobre los pensamientos propios, incluye la planeación antes de una tarea, el monitoreo durante una tarea y la autoevaluación al terminarla.

Asimismo, estudios recientes muestran su funcionalidad con estudiantes con necesidades educativas especiales (López-Escribano, & Sánchez-Montoya 2012), y se describen una serie de actividades muy interesantes para alumnado con autismo, discapacidad intelectual, motriz y visual con la que se ofrecen oportunidades de construir activamente sus conocimientos, planificar proyectos, plantear dudas y preguntas y trabajar en la resolución de problemas, todo ello desde un enfoque que les permite un aprendizaje activo y significativo.

El desarrollo de las competencias claves es un referente europeo a la hora de diseñar el currículo de los adolescentes. El empelo de Scratch puede ayudar a desarrollar descriptores de las mismas como los siguientes (Vázquez-Cano, Sevillano, & Méndez, 2011; Vázquez-Cano, & Sevillano, 2011):

- *Competencia en comunicación lingüística*: Competencia en comunicación lingüística: Una comunicación efectiva hoy en día requiere de más habilidades que simplemente leer y escribir. Con Scratch, los jóvenes aprenden a ser capaces de manipular e integrar diversos tipos de información para conseguir expresarse de forma creativa y persuasiva.

- *Tratamiento de la información y competencia digital*: Trabajar con Scratch permite al alumnado aprender a seleccionar, crear y manejar información de diverso tipoy: texto, imágenes, secuencias animadas y sonido. Al tiempo que los estudiantes adquieren experiencia trabajando con esta información, se vuelven cada vez más receptivos y críticos analizando la información que les llega del mundo que les rodea.

- *Identificación de problemas, formulación de hipótesis y solución*: Scratch permite aprender a través de un contexto significativo basado en el proceso de diseño. Crear un proyecto con Scratch requiere pensar una idea, dividir esa idea en pasos e implementar esos pasos mediante el sistema de programación de bloques del programa. Está diseñado para poder ver el resultado de la programación en el acto, por lo que los estudiantes aprenden este proceso de manera interactiva.

- *Desarrollo de las capacidades creativas y la curiosidad intelectual*: Estimula el pensamiento creativo, una habilidad muy valorada hoy en día. Scratch estimula a los estudiantes para que busquen soluciones innovadoras a problemas inesperados que surgen durante el proceso de diseño.

- *Competencia social y ciudadana*: Al ser una herramienta que permite compartir proyectos muy fácilmente, puede ser utilizada para incentivar el debate entre los jóvenes de cuestiones con importancia social, no solo en el entorno educativo sino que se puede llevar a un nivel de discusión internacional, gracias a la Comunidad Scratch.

- *Competencia para aprender a aprender*: Aprendiendo a programar con Scratch, los jóvenes van descubriendo el razonamiento crítico y el pensamiento sistemático. En sus proyectos necesitan coordinar el tiempo y las interacciones entre diferentes personajes, y su habilidad para programar esto, les proporciona una experiencia directamente relacionada con la detección de problemas, la crítica constructiva, el ensayo-error, etc. conceptos importantes dentro del pensamiento sistemático. Al trabajar en proyectos que son significativos para los jóvenes, sus propias ideas les proporcionan la motivación adecuada para sobrellevar las dificultades y retos que les plantea el proceso de diseño.

En la actualidad estas estrategias educativas de programación se están haciendo accesibles para el profesorado a través de la enseñanza masiva en abierto con cursos MOOC (Cursos en línea, masivos y en abierto) como el desarrollado por la Universitat Pompeu Fabra de Barcelona y ofertado por la plataforma Miríada titulado: "Robots y Videojuegos en las aulas: Scratch y Arduino para profesores" (https://www.miriadax.net/web/robots-videojuegos-aulas-scratch-arduino-profesores).

Figure 3. Curso MOOC programación

5. Metodología de la experiencia

La metodología de trabajo se desarrolla a través de una aula virtual creada con la herramienta de software libre eXelearning en la que se alberga una clase virtual con apuntes, videos, fotos y recursos de descarga. Los recursos creados en eXelearning son accesibles en formato XHTML o HTML5, pudiendo generarse sitios web completos (páginas web navegables), insertar contenidos interactivos (preguntas y actividades de diferentes tipos) en cada página, exportar los contenidos creados en otros formatos como ePub3 (un estándar abierto para libros electrónicos), IMS o SCORM (estándares educativos que permiten incorporar los contenidos en herramientas como Moodle), XLIFF (un estándar

para la traducción) y catalogar los contenidos con diferentes modelos de metadatos: Dublin Core, LOM, LOM-ES.

El entorno creado es accesible a través de siguiente enlace (http://desy.esy.es/TICO/OBJETOS%20DE%20APRENDIZAJE/PROG_I_VIDEOJUEGOS/qu_es_scratch.html) donde se puede ver el proyecto íntegro creado por el profesor Desiderio Ferrer Delgado para la asignatura de Informática correspondiente al primer curso de la etapa de Bachillerato.

Figure 4. Entorno para el seguimiento de Scratch en eXeLearning

La estructura pedagógica de la actividad se programó durante la tercera evaluación atendiendo a la siguiente secuencia pedagógica:

Objetivos curriculares:

1. Familiarizarse con los elementos básicos de la interfaz hombre-máquina.
2. Conocer los fundamentos físicos y lógicos de los sistemas ligados a estas tecnologías.
3. Manejar las estrategias que permiten convertir estas tecnologías en instrumentos de diseño, simulación, fabricación y control.
4. Emplear técnicas de búsqueda, elaboración y presentación de la información con criterios de realidad científica.
5. Utilizar las herramientas propias de estas tecnologías para adquirir, analizar y transformar la información, convirtiéndola en fuente de conocimiento.
6. Diseñar, actualizar y consultar la información de bases de datos relacionales.
7. Utilizar lenguajes de programación para la resolución de problemas de diferentes ámbitos, entre los que se incluyen proyectos sencillos de control.
8. Aplicar herramientas de análisis y diseño asistido por ordenador a la elaboración de un producto.
9. Discriminar qué instrumento es más adecuado para un determinado problema científico o creativo.
10. Usar los recursos informáticos como instrumento de resolución de problemas específicos.

Competencias:

a. Competencia en comunicación lingüística
b. Competencia matemática.
c. Competencia en el conocimiento y la interacción con el mundo físico.
d. Tratamiento de la información y competencia digital
e. Competencia social y ciudadana.
f. Competencia cultural y artística.
g. Competencia para aprender a aprender
h. Autonomía e iniciativa personal

Indicadores de desarrollo:

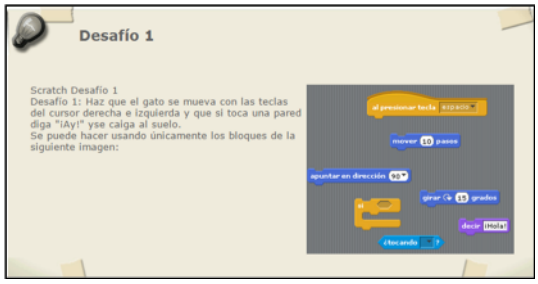
1. Conoce los elementos básicos de la aplicación Scratch.
2. Utiliza adecuadamente las herramientas de la aplicación.
3. Analiza e identifica los elementos programáticos.
4. Selecciona y utiliza los elementos de control y las funciones básicas de programación.
5. Crea en equipo un videojuego
6. Evalúa el videojuego creado por otro equipo.

Para ello, el alumnado fue progresando por tres secuencias de aprendizaje antes de llegar a la tarea final. La primera secuencia dentro del entorno virtual de aprendizaje eXeLearning consistió en la introducción al software Scratch por medio de vídeos descriptivos.

Figure 5. Introducción a Scratch

La segunda secuencia consiste en la iniciación a la programación de videojuegos por medio de tres desafíos que les proporcionan destrezas necesarias para la programación en Scratch.

Figure 6. Desafíos de programación en Scratch



Se realizaron procesos de autoevaluación, coevaluación y evaluación final de cada uno de los desafíos y de la tarea final realizada por cada grupo de estudiantes. Para orientar el procedimiento de evaluación, se diseñó una rúbrica que fue utilizada tanto por los alumnos en grupo para la coevaluación, como por el profesor para la evaluación del proyecto final (Tabla 1).

Figure 7. Rúbrica de evaluación de los proyectos con Scratch

Criterios	2 (negativa)	4 (incorrecta)	5 (correcta)	7 (perfecta)	9 (excelente)
Pantalla de presentación (5%)	No tiene pantalla de presentación	El programa no inserta diálogos que informen al usuario sobre el objetivo de la aplicación y su funcionamiento.	El programa presenta una pantalla sin animación con el manejo básico del programa.	El programa presenta una pantalla animada en la cual informa al usuario sobre el objetivo, las reglas y el manejo del juego en un formato atractivo.	Presenta una pantalla animada con el nombre del juego. Existe un botón para ver las instrucciones, y otro para empezar a jugar.
Funcionamiento (30%)	El juego apenas funciona, tiene un único nivel, no lleva cuenta de vidas ni puntos ni tiempo, no tiene pantallas de fin.	El juego solo presenta un nivel y no lleva cuenta de vidas o puntos. No presenta pantallas de Game Over ni The End	El juego presenta algún fallo, no tiene muchos niveles y el marcador o la cuenta de vidas no funcionan bien. Presenta pantallas de Game Over y The End	Funciona perfectamente, tiene varios niveles atractivos y bien diseñados y lleva marcador y vidas. Hay pantalla de Game over y The End.	Funciona perfectamente, tiene varios niveles atractivos y bien diseñados y lleva marcador y vidas. Hay pantalla de Game over y The End animada.
Programación (30%)	Apenas existe programación.	El juego tiene una programación muy básica que no demuestra gran conocimiento de las estructuras de control. Las variables no están inicializadas.	El juego muestra cierto entendimiento de las estructuras de control de programación, pero se puede optimizar haciendo mejor uso de las mismas. Existe inicialización de las variables, se parte de un estado conocido.	Se han optimizado casi todas las instrucciones y no hay repetición de comandos innecesaria, y se han usado la mayor parte de instrucciones de control. Se usan flags.	El juego incorpora para su programación una gran variedad de herramientas de control, más allá de lo aplicado en los desafíos de programación iniciales.
Jugabilidad (20%)	Es un juego conocido implementado de forma muy pobre.	Es un juego conocido pero no implementa toda la funcionalidad.	Se trata de un juego conocido implementado perfectamente	El juego se basa en un juego conocido incorporando algo nuevo.	El juego es original, muy jugable y adictivo.
Gráficos (15%)	no hay animaciones ni gráficos originales. No hay referencias ni se respetan licencias.	las animaciones son bruscas y no originales.	existen algunas animaciones pero no son originales.	Las animaciones son suaves y abundantes.	Se usan gráficos originales y las animaciones son suaves y de gran calidad. Todas las licencias son correctas y bien referenciadas

5. Resultados

La organización de la tarea final consistió en el diseño y desarrollo de un videojuego en grupo de 4 o 5 estudiantes de forma colaborativa. La actividad se desarrolló conforme a las siguientes fases:

Primera fase: Diseño de la narrativa a realizar y argumento.

Segunda fase: distribución de los roles de cada alumno/a.

Tecera fase: creación del videojuego conforme a las siguientes preguntas:

- ¿Qué queremos?
- ¿Qué objetos necesitamos crear?
- ¿Cómo interactúan entre sí?
- ¿Cuántos fondos necesitamos crear?
- ¿Cómo interactúan entre sí y con los objetos?

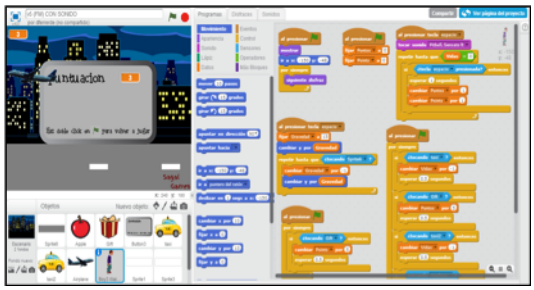
Los alumnos/as de la clase de bachillerato realizaron varios videojuegos, entre los más destacados se

encuentra el siguiente del que realizamos una descripción argumental y técnica de su creación e invitamos al lector a visualizar y jugar en el siguiente enlace:
https://scratch.mit.edu/projects/64144384/
Para iniciar el juego se tiene que pinchar la bandera verde. El muñeco salta cuando se presiona la tecla espacio y empieza la música. El objetivo del videojuego es evitar que te atropelle el taxi y evitar chocar con el resto de obstáculos. Cada vez que se juega se dispone de 3 vidas y se va obteniendo puntuación a medida que se salvan los obstáculos.

Figure 8. Juego creado por el alumnado del IES Margarita Salas (Seseña-Toledo)

El videojuego se ha diseñado con 11 objetos a los que se ha ido dando movimiento, sonido e interacción en la narrativa de juego con la programación por bloques. En la Figura 8, se puede visualizar el diseño interno del videojuego con los bloques de programación diseñados por el alumnado.

Figure 8. Entorno de programación en Scratch para el juego creado



6. Conclusiones

Las estudiantes de primero de Bachillerato implicados en esta experiencia de programación han aprendido a diseñar un proyecto de creación de un videojuego de forma colaborativa, mejorando principalmente sus competencias de expresión y creatividad. El uso de una herramienta gratuita e intuitiva como Scratch permite que el alumno desarrolle de forma interdisciplinar diferentes contenidos de distintas asignaturas y trabaje simultáneamente diversas competencias básicas desde el razonamiento crítico. En cada proyecto es necesario coordinar tiempo, interacciones entre personajes, ubicación y direccionalidad, entre otros muchos aspectos. Esto fomenta en el alumnado la destreza de afrontar situaciones problemáticas, buscar la manera de solucionarlas, el uso de la técnica ensayo-error y la crítica constructiva; conceptos que forman parte del pensamiento sistemático. Asimismo, la creación de estos proyectos fomenta el trabajo colaborativo, la coevaluación, la imaginación del alumnado y controlar, desde una mayor autonomía, su proceso y su propio aprendizaje.

Referencias

- BBC (2014). *Computer coding taught in Estonian primary schools.* Disponible en http://www.bbc.com/news/education-25648769
- Bright, G. W, (1991). Effects of Computer Programming on Cognitive Outcomes: A Meta-Analysis, Journal of Educational *Computing Research,* 7, 251-268,
- Burke, R. J., & Mattis, M. C. (2007). *Women and Minorities in Science, Technology, Engineering and Mathematics*. Elgaronline. Disponible en http://www.elgaronline.com/view/9781845428884.00026.xml
- Calder, N. (2010). Using Scratch: an integrated problem-solving approach to mathe-

matical thinking. *Australian Primary Mathematics Classroom, 15*(4), 9-14.

- Clements, D. H., & Swaminathan, S. (1995). Technology and School Change New Lamps for Old? *Childhood Education, 71*(5), 275-281.
- Code.org. (2013). What most schools don't teach. *YouTube*. Disponible en https://www.youtube.com/watch?v=nKIu9yen5n
- Deterding, S. (2012). *The Gameful Classroom. A workshop at Games*. Madison, WI: Learning & Society 8.0.
- Druin, A. (1998). *The Design of Children's Technology*. San Francisco, CA: Morgan Kaufmann Publishers.
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: a case study. *Computers & Education, 63*, 87-97.
- Gold, V. (2011). Students with disabilities, supporting literacy. *Scratched Discussions*. Disponible en http://scratched.media.mit.edu/discussions/researching-scratch/students-disabilities-supporting-literacy
- Hazzan, O., Gal-Ezer, J., & Blum, L. (2008). *A model for high school computer science education: The four key elements that make it!* Technical Symposium on Computer Science Education - SIGCSE, 281-285.
- Kapp, K. (2012). *The Gamification of Learning and Instruction*. San Francisco, CA: Pfieffer.
- Kim, H., Choi, H., Han, J., & So, H. (2012). Enhancing teachers' ICT capacity for the 21st century learning environment: three cases of teacher education in Korea. *Australasian Journal of Educational Technology (AJET), 28*(6), 965-982.
- Kordaki, M. (2012). Diverse categories of programming learning activities could be performed within Scratch. *Procedia – Social and Behavioral Sciences, 46,* 1162-1166.
- Lamb, A., & Johnson, L. (2011). Scratch: computer programming for 21st century learners. *Teacher Librarian, 38*(4), 64-68.
- LeJDD (2014). *Hamon : "Le code informatique à l'école dès septembre.* Disponible en http://www.lejdd.fr/Societe/Hamon-Le-code-informatiqu-a-l-ecole-des-septembre-675912
- López-Escribano, C., & Sánchez-Montoya, R. (2012). Scratch y necesidades educativas especiales: Programación para todos. *RED, Revista de Educación a Distancia, 34,* 1-14.
- LTS. (2010). Curriculum for Excellence. Technologies: experiences and outcomes (Govt. Rep.). Learning and Teaching Scotland (LTS). Disponible en http://www.ltscotland.org.uk/curriculumforexcellence/technologies/
- Matlin, M. W. (1993). *The psychology of women*. New York, NY: Harcort, Brace, Jovanovich.
- Nelson, J. (2009). Celebrating Scratch in libraries: creation software helps young people develop 21st-century literacy skills. *School Library Journal*, 20-21.
- Nicks, D. (19 Junio 2014). The ambitious plan to teach 100,000 poor kids to code. *Time*. Disponible en http://time.com/2901198/computer-code-van-jones-prince-yeswecode/
- NPR Staff. (25 Enero 2014). Computers are the future, but does everyone need to code? All tech considered. Disponible en http://www.npr.org/blogs/alltechconsidered/2014/01/25/266162832/computers-are-the-future-but-does-everyone-need-to-code
- Rae, L. (1996). Gender Related to Success in Science and Technology. *Journal of Technology Studies, 22*(2),21-29.
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Comm. ACM* 52(11), 60-67.
- Sánchez-Montoya, R. (2011). ¿Más avance tecnológico implica mayor inclusión? *VII Jornadas de Cooperación Educativa con Iberoamérica sobre Educación Especial e Inclusión Educativa*. Octubre, 2011, Montevideo, Uruguay.

- Scaffidi, C., & Chambers, C. (2012). Skill progression demonstrated by users in the Scratch animation environment. *International Journal of Human-Computer Interaction, 28*(6), 383-398.
- Sheldon, L. (2012). *The Multiplayer Classroom: Designing Coursework as a Game*. Boston, MA: Cengage Learning.
- Smith, A. C. (2010). Dialando: tangible programming for the novice with Scratch, processing and arduino. In *6th International Workshop on Technology for Innovation and Education in Developing Countries, Maputo, Mozambique, 21–23 January 2010*, 1-4.
- The Guardian (2012). A manifesto for teaching computer science in the 21st century. Disponible en http://www.theguardian.com/education/2012/mar/31/manifesto-teaching-ict-education-minister
- Vázquez-Cano, E., Sevillano, M.ª L. y Méndez, M.A. (2011). *Programar en Primaria y Secundaria*. Madrid: Pearson.
- Vázquez-Cano, E. y Sevillano, M.ª L (2011). *Educadores en Red. Elaboración de materiales audiovisuales para la enseñanza*. Madrid: Ediciones Académicas-UNED.
- Wartella, E. A., & Jennings, N. (2000). Children and computers: New technology-old concerns. *Children and computer technology, 10*(2), 31-43.
- Wilson, A., Hainey, T., & Connolly, T.M. (2013). Using Scratch with primary school children: an evaluation of games constructed to gauge understanding of programming concepts. *International Journal of Game-Based Learning, 3*(1), 93-109.
- Xinogalos, S. (2012). An evaluation of knowledge transfer from microworld programming to conventional programming. *Journal of Educational Computing Research, 47*(3), 251-277.
- Zichermann, G., & Cunningham, C. (2011). *Gamification by Design*. Sebastopol, CA: O'Reilly.

Forma de Citación

VÁZQUEZ-CANO, Esteban y FERRER DELGADO, Desiderio: La creación de videojuegos con Scratch en Educación Secundaria. *Revista Communication Papers*, Nº 6, páginas 63 a 73. Departamento de Filología y Comunicación de la Universidad de Girona. Recuperado el __ de_____ de 2_____ de: http://www.communicationpapers.es