

CodingFarm: videojuego como herramienta de apoyo a la docencia de conceptos de programación

Marcos Jiménez Benítez¹, Josefa Díaz-Álvarez²[0000-0003-2105-3905], and Halina Cwierz³[0000-0003-3011-487X]

¹ Universidad de Extremadura
mjimenezjju@alumnos.unex.es

² Departamento de Arquitectura de Computadores y de las Comunicaciones, Centro Universitario de Mérida, Universidad de Extremadura
mjdiaz@unex.es

³ Departamento de Ingeniería de Sistemas Informáticos y Telemáticos, Centro Universitario de Mérida, Universidad de Extremadura
hccwierz@unex.es

Abstract The field of education, including specific training in the different specialties of computer science, is looking for new tools that motivate and promote the learning of programming concepts in a gradual and simple way. The world of video games is attractive for students and presents a high potential to be used as a tool for learning the fundamental knowledge of programming, in a visual and interactive way.

This work presents a 3D videogame that places the player in a rural environment, assuming its management. To achieve its objectives, it should gradually acquire the necessary knowledge to use the tools related to terminology and intrinsic knowledge in programming.

Learning is proposed as a progressive process from the most basic concepts to the most complex ones, where the students must solve several proposed problems in order to advance in the development of the farm. The main objective is to use the videogame as a support tool in basic programming subjects, allowing students to extend the knowledge explained in class.

1. Introducción

La digitalización del sistema educativo español que se inició hace algunos años se ha visto impulsada en el último año a consecuencia de la crisis sanitaria por la COVID-19. La revolución provocada por las Tecnologías de la Información y las Comunicaciones (TIC) en toda la sociedad y, por tanto, en el ámbito educativo está siendo ampliamente estudiado [1]. La llamada cuarta revolución industrial [2] enfrenta a nuestra sociedad a cambios intensos debidos a la revolución tecnológica, donde millones de personas están conectadas utilizando dispositivos portátiles. En este contexto, aparece el paradigma de educación 4.0 [3]

que aborda los retos a los que se enfrenta la educación para adaptarse a estos cambios y aprovechar el potencial que las TIC presentan. Este proceso, además, se beneficia de la aplicación de técnicas de Inteligencia Artificial (IA) que permiten evaluar y mejorar el proceso de aprendizaje, gracias a la mejora en la recolección de datos [4].

El uso de la tecnología facilita tanto la impartición de las clases, la tutorización, así como el aprendizaje personalizado haciendo uso de herramientas alternativas. Desde este punto de vista, es innegable el peso que la industria del videojuego tiene en el mundo, incluso en una situación de pandemia, el sector ha alcanzado un crecimiento superior al de 2019 [5] con una cifra de negocio de aproximadamente 175,000 millones de dólares y situando a España como el décimo país en el mercado mundial. Dentro de la producción de los videojuegos se encuentran los juegos educativos (serious games) [6] aunando formación y entretenimiento, que suponen en España el 13,5 % de la actividad del sector del videojuego frente al 22 % a nivel mundial y cuyo principal nicho de mercado es el sector educativo (78 %). Los juegos educativos se presentan como un elemento de gran impacto en la educación [7], en gran parte debido al éxito entre la población más joven. En este contexto, los juegos educativos surgen como una gran herramienta de apoyo a la metodología de enseñanza tradicional en un campo o temática particular, sus ventajas, inconvenientes y retos han sido ampliamente analizadas [8,9].

El ámbito de las STEM (Science, Technology, Engineering and Math) no es ajeno y múltiples estudios y propuestas tienen como objetivo la introducción del videojuego como herramienta de apoyo docente en diferentes niveles educativos incluidos los estudios superiores [10,11,12,13]. Aunque el uso de videojuegos en nuestras aulas no esté completamente admitido, algunos países europeos como Eslovaquia están estudiando la posibilidad de incorporar videojuegos como Minecraft: Education Edition [14], uno de los más vendidos a nivel mundial, en sus escuelas. Sus expertos han llegado a la conclusión de que este tipo de videojuegos puede ser empleado para fomentar e infundir conocimientos y habilidades como la programación, la comunicación interpersonal, etc.

Existe una amplia variedad de videojuegos cuyo objetivo es que el usuario aprenda conceptos relacionados con la programación o lenguajes expresamente diseñados para este fin. Entre ellos se encuentran plataformas como CodeCombat¹ o CodinGame² y Scratch³. Todos ellos aúnan sencillez y un diseño artístico atractivo para los más jóvenes con el objetivo de facilitarles una primera toma de contacto con diversos lenguajes de programación como python, javascript o C++, entre otros. La principal motivación para desarrollar un videojuego propio se basa en la ventaja de tener un desarrollo adaptado y adaptable a las necesidades detectadas en los estudiantes. Así, en cualquier momento es posible actualizar y centrar el aprendizaje en una tarea de programación determinada y especificar el objetivo a conseguir para poder pasar al siguiente nivel.

¹ <https://www.codecombat.com>

² <https://www.codingame.com/>

³ <https://scratch.mit.edu>

La diferencia de la propuesta descrita frente a otras alternativas existentes, es la estrecha relación con el temario de la asignatura de Fundamentos de Programación (*FP*) impartida en el Centro Universitario de Mérida (*CUMe*). Particularmente, se destaca el aspecto de parametrización y adaptación rápida, en caso de ser necesario, que en desarrollos de terceros sería difícil de conseguir. Además, el principal interés no es el lenguaje de programación en sí sino la capacidad de pensamiento lógico, aspecto fundamental en programación.

Por todo ello, este trabajo presenta el desarrollo de un videojuego como herramienta auxiliar en el aprendizaje de las bases de la programación. El contenido y alcance del mismo está totalmente inspirado y centrado en el temario de la asignatura de FP del CUMe, perteneciente a la formación básica de los grados en Ingeniería Informática en Tecnologías de la Información (*GIITI*), Ingeniería Telemática en Telecomunicación (*GITT*) y en el plan conjunto de estudios oficiales que agrupa a las dos titulaciones.

El proyecto propuesto y denominado CodingFarm se sitúa dentro del ámbito de desarrollo de tecnologías 3D, implicando múltiples herramientas que varían desde ámbitos más técnicos, como la programación, hasta los artísticos, como son el diseño y modelado 3D. Aunque el público objetivo es el alumnado del primer curso de las titulaciones previamente mencionadas, los únicos requisitos necesarios para su utilización es leer con fluidez, tener nociones básicas de matemáticas y capacidad para analizar y resolver problemas. Esto no excluye su uso por un perfil más joven en otros niveles educativos.

Se plantean los siguientes objetivos a alcanzar en este trabajo:

- Desarrollar un videojuego 3D totalmente funcional y abierto a futuras ampliaciones.
- Incidir en el carácter de ayuda complementaria a la metodología clásica en la asignatura para afianzar la asimilación de contenidos y adquisición de competencias propias de la asignatura.
- Promover el empleo de los videojuegos educativos en el aula como una vía atractiva, moderna y viable en la enseñanza de cualquier nivel educativo.

Este documento se estructura como sigue. En primer lugar se presenta la metodología usada para el desarrollo. A continuación se describe el resultado alcanzado, donde se muestra un ejemplo de funcionamiento. La siguiente sección explica el proceso de evaluación llevada a cabo y, por último, se presentan las conclusiones obtenidas.

2. Metodología

Una de las primeras decisiones al abordar la implementación de un videojuego es el motor de juegos a utilizar. Hay dos grandes motores Unity [15] y Unreal Engine [16], que proporcionan los recursos necesarios. En este trabajo se eligió Unity ⁴, porque tiene una curva de aprendizaje más rápida, dispone de una comunidad muy activa y soporta una gran variedad de plataformas.

⁴ Versión 2019.4.10f1

Otro aspecto es la temática central que dirigirá la evolución del juego y que determinará los elementos gráficos. CodingFarm se ambienta en un pequeño poblado dedicado a la ganadería y el cultivo, donde el usuario asume el control de la gestión y debe resolver un conjunto de tareas para alcanzar la metas establecidas en las distintas secciones. La dificultad de los retos escala progresivamente de manera similar a la complejidad algorítmica vista en la asignatura en la que está basado.

La siguiente decisión es la definición de los requisitos funcionales y no funcionales. Respecto a los requisitos funcionales, se establecen los siguientes:

- Se podrá comenzar una nueva partida y reiniciar el progreso si se considera necesario.
- Se habilitará guardar el progreso en cualquier momento. Sólo se perderá en el caso de una salida forzada.
- Se dispondrá de menús informativos que faciliten la usabilidad y el control del juego.
- Los paneles informativos podrán configurarse para utilizar lenguaje técnico mediante pseudocódigo para las diferentes acciones o coloquial.
- Los objetivos a alcanzar en cada nivel estarán siempre accesibles al usuario.

Los requisitos no funcionales hacen referencia a los siguientes aspectos:

- La temática se ajusta a los contenidos de la asignatura de FP.
- El juego debe ser visualmente atractivo, apto para todos los públicos y no contener escenas explícitas o lenguaje soez.
- Se dará opción de adaptar los gráficos para mejorar el rendimiento en equipos con distintas especificaciones.
- El sistema estará adaptado para su uso en ordenadores personales (PC y laptops).
- El juego será compatible con los sistemas operativos Windows, macOS y Linux. La opción para móviles será una propuesta de mejora en el futuro.
- El juego no contendrá ningún elemento de monetización; todas sus funcionalidades podrán ser desbloqueadas mediante el avance progresivo en el mismo.

Con un desarrollo basado en niveles, cada nivel define el conjunto de tareas a completar por el jugador. Los objetivos de cada nivel se cargan desde un fichero que el jugador no puede modificar al estar incorporado dentro del proyecto. La Figura 1 muestra el esquema general de las actividades y objetivos relacionados con el desarrollo de cada nivel del juego y que pueden ser ejecutadas por los actores. La Figura 2 muestra un ejemplo de estructura de código, que puede ser asignado a un trabajador y analizado y ejecutado dentro del videojuego propuesto.

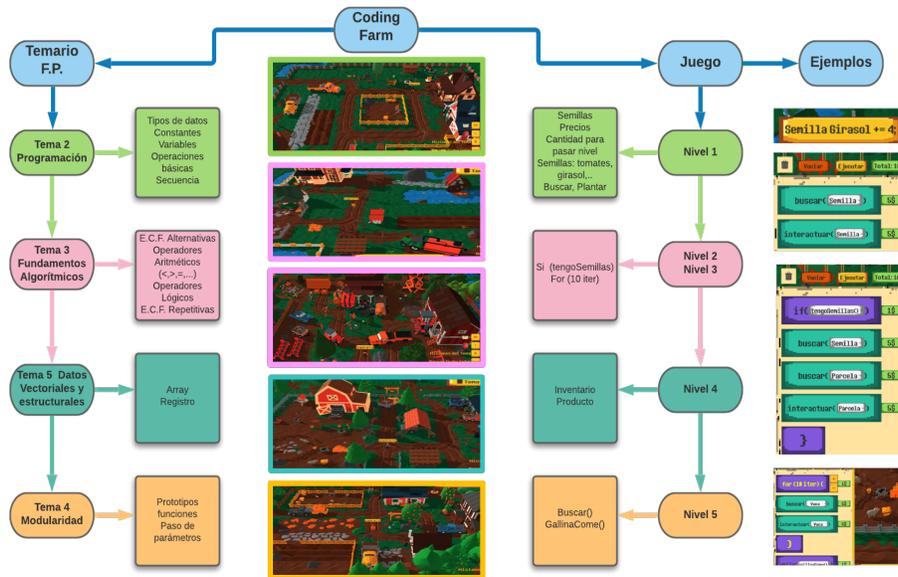


Figura 1: Esquema de actividades y objetivos generales por cada nivel del juego y relación con el temario de la asignatura.

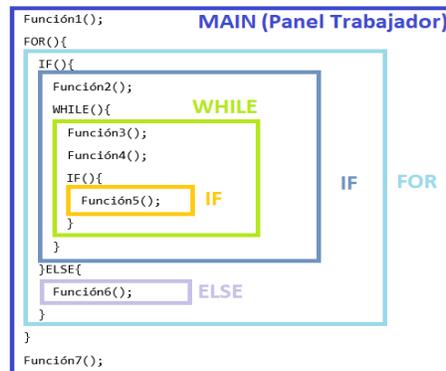


Figura 2: Ejemplo de código dividido en bloques, que puede ser analizado y ejecutado por el sistema.

Otro aspecto importante en todo desarrollo software es la fase de pruebas. Durante las diferentes iteraciones de desarrollo cinco personas se encargaron de probarlo y realizar el feedback necesario para mejorar tanto su funcionalidad como el rendimiento. Una vez finalizado, se convocó a diez estudiantes para que evaluaran el videojuego mediante un cuestionario de catorce preguntas. Los resultados se mostrarán en la sección 4.

3. Descripción de CodingFarm

El resultado es un videojuego de carácter educativo, dirigido a consolidar los conceptos básicos de programación y adquirir pensamiento lógico. Apto para todos los públicos, se ambienta en un entorno rural, colorido y visualmente atractivo, donde el concepto de economía sostenible está muy presente. El jugador se sitúa en un poblado que debe gestionar, ayudado por los aldeanos para que sus parcelas prosperen y alcancen sus objetivos.

Se presenta una única área de juego dividida en parcelas y cada una corresponde a un nivel como se observa en la parte central en la Figura 1. Al inicio, sólo el nivel 1 está activo, el resto de niveles se activan progresivamente al superar los retos asignados al nivel anterior. Cada nivel está asociado a conceptos específicos de programación. El nivel 1 aborda los conceptos de variables, constantes y operaciones asociadas; el nivel 2 introduce las estructuras de control, condicionales y las operaciones lógicas; el nivel 3 trabaja con las estructuras de control iterativas; el nivel 4 incorpora las estructuras de almacenamiento, particularmente vectores y matrices. Por último, el nivel 5 aborda la modularidad mediante el uso de funciones o procedimientos y el paso de parámetros.

Con un modelo de vista isométrica, el jugador se sitúa detrás de la cámara, toma su control y se desplaza libremente por el mapa tanto en el eje vertical (teclas **W** y **S**) como horizontal (teclas **A** y **D**), hace zoom sobre el escenario (pulsando la rueda del ratón). Se han eliminado efectos no deseados como la visualización de zonas fuera del área de jugabilidad, que interrumpen la inmersión del jugador. La velocidad de desplazamiento se incrementa al pulsar la tecla *mays* al mismo tiempo que se desplaza la cámara. La representación del cursor varía al pasar sobre elementos con los que el jugador puede interactuar.

En todo momento, la tecla **K** proporciona información de las tareas a completar en el nivel actual. Pulsando el botón derecho del ratón sobre una operación se muestra información sobre ella en formato de texto o pseudocódigo, facilitando la comprensión de los conceptos de programación. El tipo de mensaje se configura al inicio del juego o pulsando **ESC**, en cualquier nivel. En la sección 3.2 se muestra un juego de prueba relativo al nivel 1 y sus posibles opciones.

3.1. Funcionalidades

Esta sección describe brevemente las funcionalidades implementadas en el videojuego desarrollado.

- El jugador interactuará con objetos coleccionables como semillas, cubos, utensilios, etc, que desaparecen al ser recogidos.
- El jugador podrá solicitar servicios de trabajadores. El número de trabajadores se define incrementalmente por nivel con un máximo de 5 en el último. La ayuda es mediante la asignación de tareas y su uso es obligatorio en algunos niveles para alcanzar los objetivos.
- Tipo y número de tareas se incrementarán al superar niveles. Las tareas son bloques que se arrastran al panel de tareas del trabajador y permiten modificar valores de variables, recolectar objetos, modificar el flujo según una

condición, tareas repetitivas, etc. Los posibles parámetros se proporcionan como menú desplegable. Pulsando el botón derecho se accede al pseudocódigo.

- El sistema detectará errores típicos de programación en las tareas, simulando un compilador de código clásico.
- Al recoger un coleccionable, se generará otro aleatoriamente en algún lugar del área de juego.
- Cada coleccionable tendrá asociado un coste fijo y se podrá vender a los comerciantes, obtener dinero y así, liberar espacio en el inventario.
- El progreso se podrá guardar y recuperar en cualquier instante. Cualquier intento de modificación de los ficheros asociados por parte del jugador será detectado y se volverá al nivel 1.

El inventario almacena los objetos coleccionables recogidos por el jugador y contiene, entre otros, la cantidad obtenida y su valor monetario. Está accesible con la tecla **I** y tiene un tamaño limitado. En esta versión, una transacción comercial sobre un elemento del inventario afecta a la cantidad completa del elemento seleccionado.

3.2. Resultado experimental

A continuación se describe brevemente una ejecución del nivel 1 asociado al tema 1 del videojuego desarrollado. Inicialmente el jugador no dispone de capital, por lo que debe buscar y recoger todas las semillas que encuentre en el área de juego y así incrementar su inventario (Figura 3 y Figura 4). Una semilla se recoge al pulsar con el ratón sobre ella. A continuación, el jugador puede vender las semillas al granjero para obtener solvencia económica. Esta transacción se realiza pulsando con el ratón sobre el granjero y seleccionando del inventario las semillas a vender. Otra opción disponible en este nivel es el de plantar semillas y recoger la cosecha. En esta versión, ambas operaciones se realizan una a continuación de la otra.

Las tareas anteriores permiten trabajar con variables, constantes y operaciones asociadas como incrementar y decrementar, que se reflejan en el inventario y en el capital del jugador (Figura 3 nº 4). Con el dinero obtenido de la venta de las semillas podrá completar alguna de las tareas asignadas (Figura 3 nº 1), por ejemplo: “Un trabajador vaya una o más veces en busca de semilla“, esta tarea tiene un coste que se restará del capital disponible en ese momento, en caso de no tener capital no se podrá completar la ejecución.

Las tareas se asignan a un trabajador, la tecla *espacio* localiza al trabajador del nivel si no se encuentra a la vista. A la asignación de tareas se accede pulsando con el ratón en el trabajador, a la derecha aparecerá la ventana de tareas disponibles; a la izquierda, las tareas asignadas al trabajador. Basado en bloques (tipo *scratch*), las tareas se asignan arrastrándolas desde la ventana derecha a la izquierda. En la parte superior izquierda se muestran las operaciones que se pueden realizar sobre el panel izquierdo y el coste de las tareas incluidas en él, junto a información de ayuda; la parte central inferior mostrará información de

la última operación realizada en formato pseudocódigo o texto; la parte superior derecha especifica el dinero acumulado por el jugador (Figura 4 n^o 6). En el panel de requerimientos para superar el nivel se detallará el progreso del jugador. Así, se especificará por cada uno el número de tareas completadas sobre el total de requeridas (Figura 4 n^o 8). Mientras no se haya completado aparecerá en rojo, una vez completada una tarea, ésta cambiará a verde. Cada nivel incorpora mayores requerimientos a superar, nuevos elementos con los que interactuar, pero también se incrementa el número de trabajadores a los que asignar tareas. Los trabajadores disponibles podrán estar realizando sus tareas en paralelo ⁵.

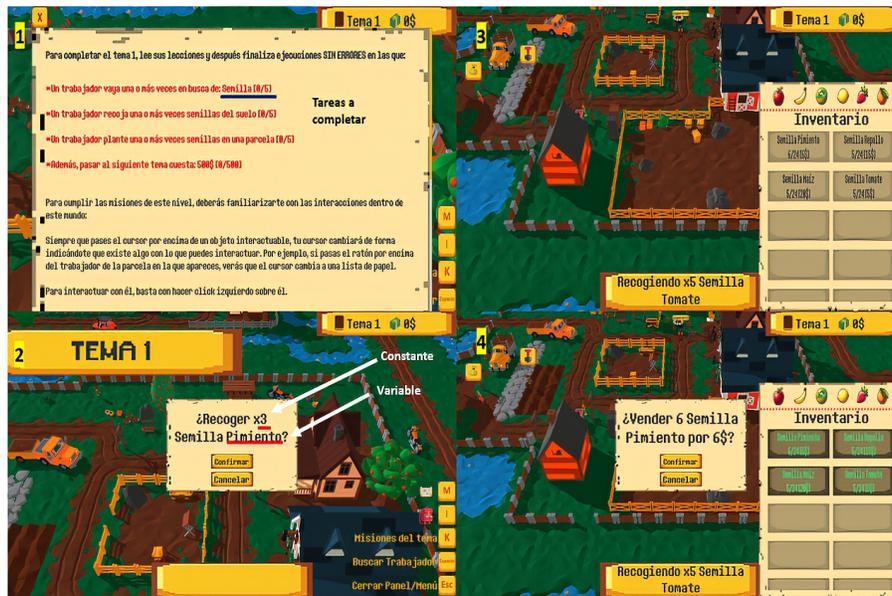


Figura 3: Primera parte: pasos a seguir para completar una ejecución de la primera tarea tema 1.

4. Evaluación de CodingFarm

La evaluación de CodingFarm se ha realizado durante la fase de desarrollo y una vez completada la implementación. El principal objetivo durante la fase de desarrollo era la detección de posibles errores para ser corregidos. La versión completa pasó de nuevo por este proceso para garantizar una implementación sin errores y, de forma paralela, evaluar el rendimiento de la aplicación. A continuación, se explican los resultados obtenidos durante los procesos de evaluación.

⁵ Acceso a video demostrativo:
<https://zurbaran.unex.es/cloud/index.php/s/GiYqe2Ay852W6F3>



Figura 4: Segunda parte: pasos a seguir para completar una ejecución de la primera tarea tema 1.

4.1. Evaluación de rendimiento

Como se mencionó en la sección 2 varias personas testearon el juego en diversos equipos para detectar posibles errores y problemas de rendimiento. Se seleccionó a tres estudiantes de último curso de GIITI, que habitualmente son usuarios de videojuegos y a dos estudiantes sin conocimientos de Informática. La retroalimentación recibida permitió corregir los problemas detectados y mejorar el rendimiento de la aplicación en diferentes equipos. Para mejorar los problemas de rendimiento se utilizaron, entre otros, recursos como una clase gestor base siguiendo el patrón *singleton*, reubicar objetos en lugar de destruir/crear continuamente, usar el canal de renderizado universal (URP), ajustar la tasa de refresco o el trazado de sombras.

La Figura 5 muestra los resultados conseguidos en función del rendimiento obtenido en un conjunto de equipos con diferentes características. Se puede apreciar, que el rendimiento se ajusta a los estándares de la industria, cumpliendo así con el objetivo de desarrollar una solución *software* útil a la par que eficiente.

4.2. Satisfacción del usuario

La versión definitiva de CodingFarm se evaluó también desde el punto de vista de satisfacción del usuario. Se convocó a diez estudiante, ocho del CUME (4 de primer curso y 4 de tercer/cuarto curso) y dos estudiantes de titulaciones no tecnológicas. Los estudiantes jugaron dos horas y posteriormente se les instó a contestar un formulario con 14 preguntas (ver Tabla 1). La respuesta a cada

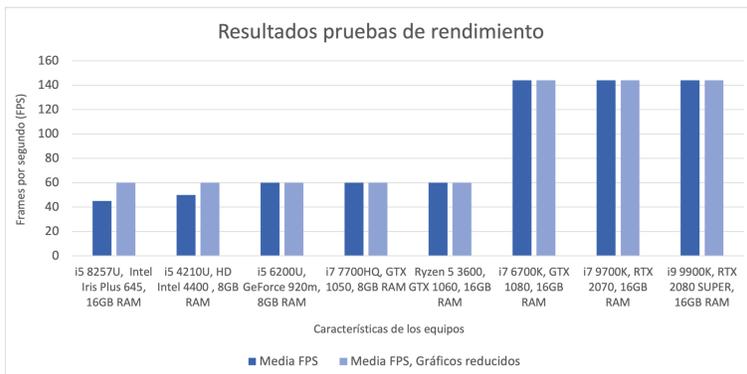


Figura 5: Resultados de rendimiento obtenidos del testeo. El eje X representa las características de los equipos, el eje Y el número de frames por segundo (FPS) y las barras las mediciones tanto para la media con calidad normal y reducida. Mayor FPS es mejor.

pregunta es un nivel de satisfacción que va desde el valor *Muy bajo* (1) hasta *Muy alto* (5). Los estudiantes del CUMe alcanzaron el nivel 4 mientras que los 2 de otras titulaciones se quedaron en el nivel 2 y todos indicaron comprender los conceptos abordados. Además, los de primer curso indicaron que les ayudaba a aclarar algunas dudas y los pertenecientes a otras titulaciones mostraban su satisfacción al entender la lógica de las operaciones.

La Figura 6 muestra el resultado obtenido, donde se observa que los participantes valoran positivamente los aspectos de CodingFarm evaluados. Así, la mayoría lo califica como adecuado, bueno o muy bueno, siendo esta última valoración la predominante en 10 de las 14 preguntas. Solo los aspectos relativos a la presentación de objetivos y ayuda iniciales (preguntas 2 y 6), muestran un grado de satisfacción bajo y a mejorar en futuras actualizaciones.

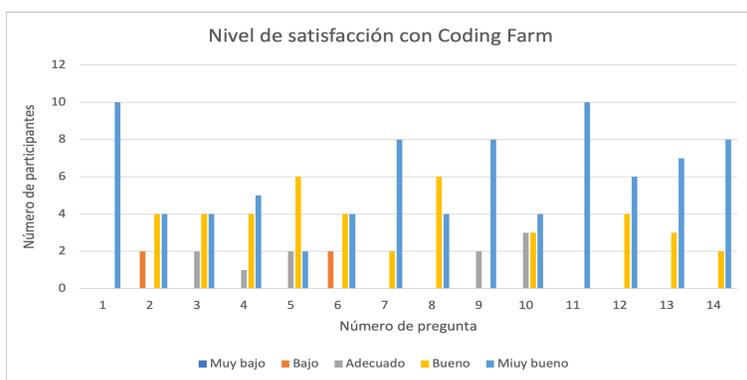


Figura 6: Resultado de evaluación de CodingFarm. El eje X representa las preguntas realizadas, el eje Y el número de participantes en la evaluación y cada barra de un color el nivel de satisfacción para la pregunta dada desde 1 (muy bajo) hasta 5 (muy alto).

Preguntas

-
1. ¿Cómo le ha resultado la instalación del juego?.
 2. La introducción del juego es clara y ayuda al jugador a entender su funcionamiento.
 3. El juego proporciona contenido que estimula mi atención.
 4. Percibo que la mayoría de las actividades están orientadas al aprendizaje.
 5. No me distraigo de las tareas en las que el jugador debe estar concentrado.
 6. Los objetivos del juego se presentan claramente al inicio del juego.
 7. Los objetivos de cada nivel están claramente presentados.
 8. Entiendo los objetivos de aprendizaje del juego.
 9. Los retos son adecuados al nivel del primer curso.
 10. El juego suministra hitos que ayudan a superar los retos.
 11. La dificultad de los retos se incrementa a la vez que mis habilidades para superarlos.
 12. Soy capaz de capturar las ideas básicas del objetivo de aprendizaje.
 13. El juego motiva al jugador a integrar el conocimiento aprendido.
 14. ¿Recomendarías aplicar videojuegos a objetivos particulares en otras asignaturas?
-

Tabla 1: Cuestionario de evaluación de satisfacción del usuario para Coding Farm.

Adicionalmente, el próximo curso se evaluará la eficiencia del videojuego en la asignatura de *FP*. Se propondrá a un grupo de estudiantes probar el videojuego a medida que se impartan los temas. Al finalizar la asignatura, se comprobarán los resultados obtenidos por el grupo experimental y el de control.

5. Conclusiones

En este trabajo, se ha presentado un videojuego educativo abierto y parametrizable, orientado a servir como herramienta docente de apoyo en el aprendizaje de conceptos básicos de programación. El objetivo principal es que el estudiante pueda reforzar los conceptos adquiridos en las clases, de forma individualizada e independiente. Con una estructura basada en niveles de dificultad progresiva, el videojuego presenta un entorno visualmente atractivo y proporciona ayuda visual tanto en formato texto como pseudocódigo para mejorar la inmersión del jugador y favorecer el aprendizaje.

Es importante destacar el trabajo realizado para la obtención de un rendimiento adecuado que se ha conseguido evitando sobrecargar con operaciones software para destruir y crear objetos de forma continua, realización de ajustes gráficos en cuanto a renderizado, sincronización, ajuste de sombras y sistema de iluminación. La mayoría de los aspectos ha recibido una valoración muy positiva por parte de los participantes en la sesión de evaluación tanto en el rendimiento como en los aspectos relacionados con los objetivos de conceptos de programación y jugabilidad.

Aunque, el contenido del mismo está totalmente inspirado en el temario de la asignatura de Fundamentos de Programación, impartida en el Centro Universitario de Mérida, su carácter de formación básica lo hace adecuado no solo para estudiantes de las diferentes especialidades de Ingeniería Informática sino para cualquier persona interesada en adentrarse en el mundo de la programación.

Agradecimientos

Agradecemos el apoyo del Ministerio de Economía y Competitividad de España en el marco del proyecto TIN2017-85727-C4-{2,4}-P, Agencia Estatal de Investigación en el marco del proyecto PID2020-115570GB-C21, de la Junta de Extremadura, proyecto GR15068.

Referencias

1. Ronald M. Hernandez. Impacto de las tic en la educación. *Dialnet*, 5(1):325–347, 2017.
2. Klaus Schwab. *The Fourth Industrial Revolution*. Penguin Books, 2017.
3. A. Aziz Hussin. Education 4.0 made simple: Ideas for teaching. *International Journal of Education and Literacy Studies*, 6(3):92–98, 2018.
4. Monica Ciolacu, Ali Fallah Tehrani, Rick Beer, and Heribert Popp. Education 4.0 — fostering student’s performance with machine learning methods. In *2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME)*, pages 438–443, 2017.
5. Libro Blanco del desarrollo español de videojuegos 2020. <https://www.dev.org.es/publicaciones/libro-blanco-dev-2020>, 2021.
6. Ralf Dörner, Stefan Göbel, Wolfgang Effelsberg, and et al. *Serious Games, Foundations, Concepts and Practice*. Springer, 2016.
7. F. Almeida and J. Simoes. The role of serious games, gamification and industry 4.0 tools in the education 4.0 paradigm. *Contemporary Educational Technology*, 10:120–136, 2019.
8. Cristina Alonso-Fernandez, Antonio Calvo, Manuel Freire, and et al. Systematizing game learning analytics for serious games. In *2017 IEEE Global Engineering Education Conference (EDUCON)*, pages 1111–1118, 2017.
9. Richard E. Mayer. Computer games in education. *Annual Review of Psychology*, 70(1):531–549, 2019.
10. Leonard A. Annetta, Wendy M. Frazier, Elizabeth Folta, and et al. Science teacher efficacy and extrinsic factors toward professional development using video games in a design-based research model: The next generation of stem learning. *Journal of Science Education and Technology*, pages 47–61, 2013.
11. Robert Ormsby, Robin Daniel, and Marka Ormsby. Preparing for the future with games for learning: Using video games and simulations to engage students in science, technology, engineering, and math. *Astropolitics*, 9(2-3):150–164, 2011.
12. A. Jamaludin and D. Hung. Problem-solving for stem learning: navigating games as narrativized problem spaces for 21st century competencies. *Research and Practice in Technology Enhanced Learning*, 12(1), 2017.
13. Luis Fernando Calvo, Raúl Herrero Martínez, and Sergio Paniagua-Bermejo. Influencia de procesos de ludificación en entornos de aprendizaje stem para alumnos de educación superior. *Trilogía Ciencia Tecnología Sociedad*, 12(22), 2020.
14. Jakub Šajben, Nika Klimová, and Gabriela Lovászová. Minecraft: Education edition as a game-based learning in slovakia. In *Minecraft: Education Edition as a Game-Based Learning in Slovakia*, 07 2020.
15. <https://unity.com>. Última consulta: 24-11-2020.
16. <https://www.unrealengine.com>. Última consulta: 24-11-2020.