

De Jugadores a Programadores: Un Prototipo de Juego Serio como Herramienta Educativa

From gamers to programmers: a serious game prototype as an educational tool

Oscar Zatarain Aguirre¹, Abraham López Nájera², Alejandra Mendoza Carreón³, Cynthia Vanessa Esquivel Rivera⁴ y René Noriega Armendáriz⁵

Universidad Autónoma de Ciudad Juárez, Instituto de Ingeniería y Tecnología, Av. Del Charro #450 norte, Colonia Partido Romero, C.P. 32310, Ciudad Juárez, Chih. México

¹al132279@alumnos.uacj.mx, ²abraham.najera@uacj.mx, ³alemendo@uacj.mx, ⁴cyesquiv@uacj.mx, ⁵rnoriega@uacj.mx

Fecha de recepción: 20 de septiembre de 2024

Fecha de aceptación: 28 de marzo de 2025

Resumen. Las industrias requieren cada vez más que las personas tengan conocimientos de programación. El objetivo fue desarrollar un prototipo de juego que introduzca a los jóvenes en la programación utilizando líneas de código de una forma lúdica. Se utilizó la metodología de prototipos ya que se adaptaba mejor en la planeación, desarrollo y evaluación del proyecto. Las escuelas tienen problemas para introducir a los estudiantes a la programación ya que es un tema complejo de integrar al plan educativo sin las herramientas adecuadas, y de igual manera, es difícil de aprender por cuenta propia para cualquier joven interesado. Esto ha sido un problema para las industrias ya que hace falta talento en el área de informática. Para poder asistir a las instituciones en este problema se ha desarrollado un juego serio que facilita a los estudiantes la introducción a la programación brindando la oportunidad de practicar de una forma entretenida.

Palabras clave: Programación, Juegos serios y Educación.

Summary. Industries increasingly require programming skills from individuals. The objective was to develop a game prototype that would introduce young people to programming using lines of code in a playful way. The prototyping methodology was used because it was better suited for project planning, development, and evaluation. Schools struggle to introduce students to programming because it is a complex subject to integrate into the curriculum without the appropriate tools, and it is also difficult for any interested young person to learn independently. This has been a problem for industries because there is a lack of talent in the computer science field. To assist institutions with this challenge, a serious game has been developed that facilitates students' introduction to programming by providing the opportunity to practice in an entertaining way.

Keywords: Programming, Serious Games and Education.

1 Introducción

La tecnología se integra cada vez más en el funcionamiento de las industrias y esto genera una creciente demanda de profesionales con conocimientos en programación. Desarrollar habilidades de pensamiento computacional se ha vuelto esencial, ya que permite a las personas resolver problemas de manera lógica y eficiente, habilidades altamente valoradas en el entorno laboral actual. Con el progreso de la tecnología, su accesibilidad también aumenta, permitiendo a los jóvenes familiarizarse más con las computadoras desde una edad temprana. Las instituciones educativas pueden aprovechar esta familiaridad para fomentar estas habilidades mediante la implementación de nuevas técnicas de enseñanza, como la gamificación y los juegos serios.

Para desarrollar este tipo de software, existe una amplia variedad de herramientas disponibles, que incluyen programas para la creación de gráficos 2D y 3D, motores de juego y software para la generación de música. Este documento presenta la creación de un prototipo de juego serio desarrollado bajo la metodología de prototipos para el desarrollo de software, la cual permite iterar rápidamente, obtener retroalimentación temprana y realizar ajustes continuos, asegurando que el producto final sea funcional y esté alineado con las necesidades de los usuarios. Además, se incluyen pruebas realizadas con estudiantes de nivel medio superior para evaluar la usabilidad del prototipo en la introducción de los jóvenes a la programación, utilizando computadoras con acceso a internet y probando en los tres navegadores más comunes.

2 Estado del Arte

Al comenzar a programar a una edad temprana, los estudiantes se familiarizan con conceptos tecnológicos clave para entender el entorno digital actual, lo que resulta altamente beneficioso, ya que no solo les permite desarrollar habilidades esenciales como la resolución de problemas y el pensamiento computacional, sino que también puede estimular su interés en las ciencias de la computación. Este enfoque no solo mejora su interacción con la tecnología, sino que también los prepara para un mercado laboral cada vez más orientado hacia lo digital, permitiéndoles explorar y crear con tecnología desde el principio [1]. Sin embargo, en el campo de la programación, la demanda de talento supera significativamente la oferta. Un estudio de la Oficina de Estadísticas Laborales de Estados Unidos predijo que para el año 2020 existirían 1.4 millones de empleos en el área de ciencias computacionales, pero señaló que solo habría 400,000 graduados disponibles para cubrir esos puestos, lo que evidencia una brecha considerable en la fuerza laboral calificada [2].

La gamificación ha demostrado ser una herramienta efectiva en la enseñanza de la programación, ya que aumenta la motivación y la participación de los estudiantes mediante el uso de puntos, insignias y retos interactivos. Estudios han mostrado que técnicas gamificadas mejoran la comprensión y el rendimiento en cursos de programación, al hacer el aprendizaje más atractivo y accesible para los alumnos. Sin embargo, también se destacan algunos desafíos, como la necesidad de un diseño cuidadoso para evitar que la gamificación se convierta en una distracción en lugar de una ayuda educativa [3].

Existen varios diseños de juegos que se han probado para facilitar el aprendizaje de los jóvenes en esta área. Algunos de estos diseños requieren que los jóvenes escriban líneas de código o utilicen bloques para familiarizarse con el concepto de programación. A continuación, se mencionan algunos.

CodeCombat es una plataforma educativa que enseña programación mediante juegos interactivos. Los usuarios resuelven desafíos de codificación para controlar personajes y completar tareas en un entorno de fantasía. La plataforma admite lenguajes como Python y JavaScript, haciendo el aprendizaje de la programación atractivo al integrar ejercicios en un formato de juego [4].

Tok-program es un software para la enseñanza de programación que está basado en la interacción tangible que incluye actividades de aprendizaje relacionadas con diseño algorítmico, introducción a la programación, control de flujo, organización de datos y modularidad [5].

Karel the Dog es una herramienta educativa para enseñar programación mediante un entorno interactivo. En el curso CodeHS, Karel, un perro robot, realiza tareas en una cuadrícula. Los estudiantes escriben código para guiar a Karel en la navegación, recolección de objetos y resolución de acertijos, aprendiendo conceptos fundamentales como ciclos, condicionales y funciones de manera práctica [6].

Scratch fue desarrollado por el MIT, y es un lenguaje de programación visual donde los usuarios crean historias, juegos y animaciones interactivas utilizando un sistema de codificación basado en bloques. Está diseñado para que niños y principiantes aprendan conceptos de programación de una forma creativa y lúdica [7].

CSS Grid Garden es un juego en línea que enseña técnicas de diseño de CSS Grid. Los jugadores resuelven desafíos de codificación para posicionar y diseñar elementos en un jardín virtual, recibiendo retroalimentación inmediata. Es una manera divertida y práctica de aprender CSS Grid [8].

3 Metodología

Para desarrollar el prototipo de juego, se empleó la metodología de desarrollo por prototipos [9], que facilita la planificación y creación rápida de diseños. Se llevaron a cabo tres ciclos: diseño de interfaz y menús, creación de mecánicas de juego, y mecánicas avanzadas, y en cada ciclo se incluía la obtención de retroalimentación para refinar el prototipo. El juego permite a los usuarios escribir código para controlar acciones y progresar en los niveles, ofreciendo así una introducción a la programación.

3.1 Primer ciclo - Diseño de interfaz y menús

Para el desarrollo del software, los requisitos mínimos del equipo incluían 4 GB de RAM, un procesador Intel Core i7-3770, 100 GB libres en el disco duro y Windows 10. La interfaz del prototipo debía permitir una navegación clara con textos descriptivos, una disposición intuitiva de los elementos y un esquema de colores agradable. Además, los botones debían indicar visual y auditivamente su activación.

El prototipo se desarrolló en Unity 3D, que ofrece herramientas para crear menús e interfaces. La interfaz incluye áreas para jugar, escribir, visualizar y ejecutar líneas de código, y opciones del juego, todo diseñado con una temática de magia y fantasía, como se muestra en la Figura 1.



Figura 1. Interfaz de usuario.

Se evaluó el prototipo con 16 voluntarios, identificando mejoras como reubicar el botón para mostrar comandos y añadir un botón de cierre para el menú, que inicialmente obstruía la vista del juego. Las mecánicas del juego incluían la introducción y reconocimiento de código, asistencia en errores, reproducción de música de ambiente y un tutorial sobre las mecánicas.

Las mecánicas permiten al jugador controlar al personaje mediante líneas de código predefinidas, moviéndose en cuatro direcciones arriba, abajo, izquierda y derecha, e interactuando con el entorno. El nivel se completa al cumplir los objetivos, como llegar a la salida o recolectar objetos, con un menú de texto que guía al jugador en el uso de comandos y la progresión a los siguientes niveles.

3.2 Segundo ciclo - Creación de las mecánicas de juego

Se construyó el prototipo usando diseño rápido e implementando las mecánicas del juego en Unity, exportado a un servidor web para pruebas. Un tutorial guiado por el personaje principal enseña las mecánicas del juego, como se muestra en la Figura 2. El reconocimiento de comandos se realiza en un campo de texto que divide y compara las instrucciones con una lista interna, marcando las correctas en verde y las incorrectas en rojo.



Figura 2. Prototipo del tutorial de juego.

El movimiento del personaje se gestiona mediante nodos que definen las direcciones disponibles, esto se muestra en la Figura 3. El objetivo del nivel es obtener la llave y llegar a la puerta; al completarlo, el usuario avanza al siguiente nivel.

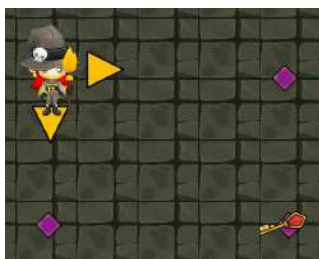


Figura 3. Prototipo de la mecánica de movimiento.

En una segunda sesión de pruebas, se identificaron errores en el diseño y se ajustó la mecánica de introducción de código. La retroalimentación indicó que el tutorial no era lo suficientemente claro, por lo que se

mejoró con imágenes explicativas. Además, se añadió un sistema que muestra el camino hacia el siguiente nodo disponible y se actualizó el tutorial con esta información.

3.3 Tercer ciclo - Mecánicas avanzadas

Para el tercer ciclo, se incluyeron mecánicas avanzadas como mostrar los puntos de vida del jugador, una pantalla de derrota, enemigos que dañan al jugador y otorgan maná al ser vencidos, y la capacidad de atacar y usar poderes especiales. Al finalizar una partida, el sistema califica al jugador y ofrece opciones para rehacer o avanzar de nivel, además permite usar un comando para repetir acciones.

Estas mecánicas añaden complejidad al juego con recursos adicionales y obstáculos. Los puntos de vida, maná y la disponibilidad del poder más fuerte se muestran mediante imágenes en la pantalla. El jugador puede atacar enemigos a corta distancia, ganar puntos para hechizos adicionales, y se muestra una pantalla de derrota si pierde todos los puntos de vida.

Se incluye una pantalla de selección de niveles para visualizar el progreso y la puntuación de cada nivel. El jugador inicia con 3 puntos de vida y pierde la partida si llegan a cero. Los enemigos son el principal obstáculo y se eliminan con el comando de ataque, que también otorga puntos de maná, la Figura 4 muestra la eliminación de un enemigo.

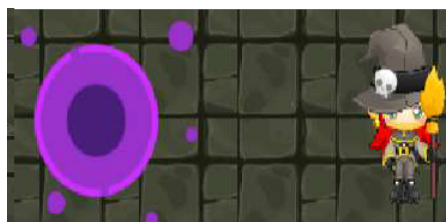


Figura 4. Enemigo siendo derrotado.

El sistema de calificación asigna entre una y tres estrellas según el número de movimientos utilizados en el nivel. Se añadió un menú de derrota para reiniciar el nivel o elegir otro, y un menú de selección de niveles para elegir desafíos y ver el progreso. Se implementó un comando para repetir acciones y un botón para eliminar comandos rápidamente. Además, se modificó el sistema de calificación para no contar las repeticiones de comandos, y se añadió una mecánica para pasar por distintos puntos múltiples veces para cumplir objetivos, como lo muestra la Figura 5.

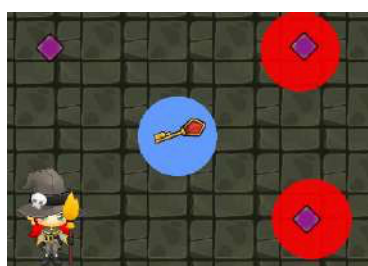


Figura 5. Puntos por los que el jugador tiene que pasar para obtener una llave.

4 Resultados experimentales

Durante las pruebas se recibieron 35 alumnos de secundaria y preparatoria que probaron el software y contestaron una encuesta. Esta se realizó en Google Forms, ya que permite crear cuestionarios para que los usuarios proporcionen respuestas y recolectar la información. Las respuestas a las preguntas van de acuerdo con la escala de Likert teniendo una escala de 5 opciones, Totalmente en Desacuerdo como la cantidad mínima y Totalmente de Acuerdo como la cantidad máxima. A continuación, se muestran los resultados obtenidos; las preguntas van en relación con la usabilidad y utilidad del juego.

4.1 Resultados de Usabilidad

La Figura 6a muestra la gráfica obtenida con la información a la pregunta ¿Los elementos de la interfaz comunican bien su función? Teniendo como mayores resultados De acuerdo con un 65% y Totalmente de acuerdo con 28%. La Figura 6b son los resultados obtenidos de la pregunta ¿El tutorial es claro al explicar las funciones del juego? Donde los resultados en su mayoría fueron 45% para Totalmente de acuerdo y De acuerdo.



Figura 6a. Pregunta 1 Usabilidad



Figura 6b. Pregunta 2 Usabilidad

La pregunta ¿Los comandos son claros en su función? Aparejada en la Figura 7a muestra los que los resultados de mayor obtención son De acuerdo con 48% y Totalmente de acuerdo con 45%, mientras que la Figura 7b muestra los resultados de la pregunta ¿Es claro lo que se tiene que hacer para pasar los niveles? Donde los de mayor relevancia son De acuerdo con 60% y Totalmente de acuerdo con 37%

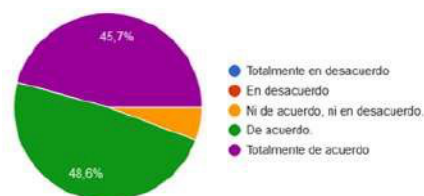


Figura 7a. Pregunta 3 Usabilidad

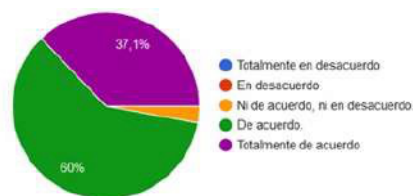


Figura 7b. Pregunta 4 Usabilidad

Los resultados de la pregunta ¿El nivel de introducción deja en claro cómo se maneja el juego? Obteniendo como mayores resultados De acuerdo con 62% y Totalmente de acuerdo con 34%, estos se muestran en la Figura 8a. La Figura 8b muestra los resultados de mayor relevancia con un 48% en Totalmente de acuerdo y un 42% en De acuerdo hacia la pregunta ¿El juego es divertido?

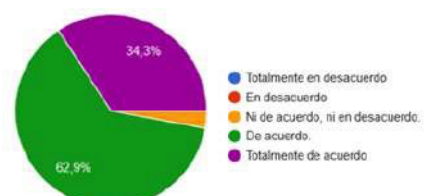


Figura 8a. Pregunta 5 Usabilidad



Figura 8b. Pregunta 6 Usabilidad

Por último, la Figura 9 muestra los resultados de la pregunta ¿El juego es muy complicado?, donde la mayor parte de los resultados fueron Ni de acuerdo, ni en desacuerdo con 40%, En desacuerdo con 28% y Totalmente en desacuerdo con 17%

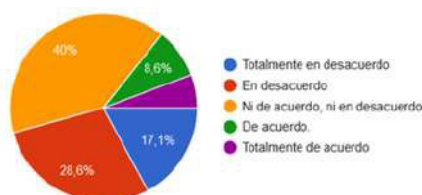


Figura 9. Pregunta 7 Usabilidad

4.2 Resultados de Utilidad

En la Figura 10a se muestra la gráfica con los resultados de la pregunta de utilidad ¿Considera que el juego le sirvió para aprender programación básica? Donde los de mayor incidencia son De acuerdo con 62% y Totalmente de acuerdo con 28%. Los resultados de la pregunta ¿Después de haber usado este juego serio ha adquirido interés en el tema de programación? Fueron 51% De acuerdo y 37% Totalmente de acuerdo como de mayor relevancia y se muestran en la Figura 10b.

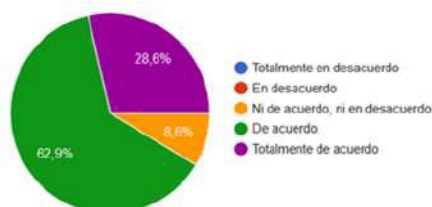


Figura 10a. Pregunta 1 Utilidad



Figura 10b. Pregunta 2 Utilidad

Para la pregunta ¿Después de haber usado este juego ha adquirido un mejor entendimiento sobre la programación? Donde los resultados se muestran en la Figura 11a, con mayor impacto fueron 51% De acuerdo y 31% Totalmente de acuerdo, mientras la Figura 11b muestra los resultados de la pregunta ¿Considera que el uso de juegos serios puede ser beneficioso para los alumnos en un aula de clase? Donde el impacto estuvo cerca en resultados obteniendo 37% Totalmente de acuerdo, 34% De acuerdo y 28% Ni de acuerdo, ni en desacuerdo.

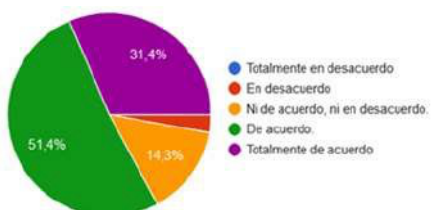


Figura 11a. Pregunta 3 Utilidad



Figura 11b. Pregunta 4 Utilidad

Por último, los resultados de la pregunta ¿Considera que el uso de juegos serios puede ser beneficioso para el auto aprendizaje? Fueron 42% De acuerdo, 31% Totalmente de acuerdo y 25% Ni de acuerdo, ni en desacuerdo, los cuales se muestran en la Figura 12.



Figura 12. Pregunta 5 Utilidad

4.3 Discusiones

Los resultados de las pruebas de usabilidad del software fueron en su mayoría positivos. La estética del juego fue bien recibida por la mayoría de los usuarios. Aunque no se registraron muchos problemas con la comprensión de las funciones de los elementos, algunos estudiantes encontraron que el uso de colores dificultaba la comprensión de la interfaz, sugiriendo que una paleta de colores más variada podría ayudar a diferenciar mejor los elementos en pantalla.

El tutorial fue eficaz en explicar las mecánicas del juego, pero algunos usuarios encontraron las mecánicas complicadas. Esto podría deberse a que el juego es muy estricto en la forma en que reconoce los comandos introducidos por el jugador.

En cuanto a la utilidad, los resultados fueron positivos, con los jóvenes indicando una buena comprensión de la programación y creyendo que los juegos serios pueden ser efectivos para la educación. Investigaciones futuras deberían considerar simplificar las mecánicas del juego para evitar confundir a los estudiantes y priorizar el desarrollo de un tutorial, dado que este sistema es crucial para enseñar a los usuarios a utilizar el software.

5 Conclusiones y Trabajo Futuro

A medida que la tecnología avanza, aumenta la demanda de conocimientos en programación, tanto en el ámbito laboral como en proyectos personales. Para facilitar el aprendizaje de los jóvenes, se ha desarrollado un prototipo de juego serio para introducir conceptos de programación mediante líneas de código. El juego incluye escenarios que utilizan estructuras básicas de control y un tutorial para enseñar las mecánicas. Se creó un área para que los jugadores introduzcan comandos, con un sistema que marca en verde las instrucciones reconocidas y en rojo las no reconocidas, permitiendo visualizar el orden de ejecución. Integrar las enseñanzas en las mecánicas del juego mejora los resultados educativos y mantiene el interés al hacerlo divertido. Las evaluaciones del prototipo fueron positivas, destacando la estética, la música y la claridad del juego, además de que los estudiantes reportaron una mejor comprensión de la programación y consideraron beneficiosos los juegos serios para el aprendizaje en el aula y de forma autónoma.

Referencias

- [1] E. Ferrari, “Teaching the principles of computer science early in K–12 schools,” EdTech Magazine (Junio, 2022). [En línea]. Disponible en: <https://edtechmagazine.com/k12/article/20>. [Último acceso: febrero 17, 2024].
- [2] School of Computer Science, Carnegie Mellon University, “Meeting the demand,” Carnegie Mellon University (n.d). [En línea]. Available: <https://www.cs.cmu.edu/link/meeting-demand>. [Último acceso: Febrero 20, 2024].
- [3] Zehui Zhan, Luyao He, Yao Tong, Xinya Liang, Shihao Guo, Xixin Lan, The effectiveness of gamification in programming education: Evidence from a meta-analysis, Computers and Education: Artificial Intelligence, Volume 3, 2022, ISSN 2666-920X, Disponible en: <https://doi.org/10.1016/j.caeai.2022.100096>.
- [4] CodeCombat, “CodeCombat,” CodeCombat (n. d.). [En Línea]. Disponible en: <https://codecombat.com/>. [Último acceso: Marzo 13, 2024].
- [5] Serrano Herrera R.A., Joaquín Salas C., Gutiérrez Marcos R., «Tok-Program: software basado en interacción tangible para la enseñanza de programación,» en Futuro Digital: Avances y paradigmas tecnológicos, Editorial AlfaOmega, ISBN: 978-607-538-708-6, ANIEI, México 2020, pp. 237-242.
- [6] CodeHS, “Introduction to Computer Science,” CodeHS (n. d.). [En Línea]. Disponible en: <https://codehs.com/course/719/overview>. [Último acceso: Marzo 10, 2024].
- [7] MIT Media Lab, “Scratch,” MIT Media Lab (n. d.). [En línea]. Disponible en: <https://scratch.mit.edu/>. [Último acceso: Marzo 10, 2024].
- [8] D. Hara, “CSS Grid Garden,” CSS Grid Garden (n. d.). [En línea]. Disponible en: <https://cssgridgarden.com/>. [Último acceso: Marzo 10, 2024].
- [9] Ecured, “Modelo de prototipos,” Ecured (n. d.). [En línea]. Disponible en: https://www.ecured.cu/Modelo_de_prototipos. [Último acceso: Marzo 14, 2024].