**Enigma del Tiempo**

Federico Robles, Andrew González, Jesús Bone

Desarrollo Basado en Plataformas – Arquitectura y Plataformas de Servidores, PUCESE

Instructor: PhD. Jaime Sayago Heredia.

[jaime.sayago@pucese.edu.ec](mailto:jaime.sayago@pucese.edu.ec)

[firobles@pucese.edu.ec](mailto:firobles@pucese.edu.ec), [afgonzales@pucese.edu.ec](mailto:afgonzales@pucese.edu.ec), [jjbone@pucese.edu.ec](mailto:jjbone@pucese.edu.ec).

**Resumen:** El videojuego Enigma del Tiempo es un proyecto basado en el motor gráfico de Unity, en el cual a partir de diseño gráfico y codificaciones se logró realizar los diversos diálogos y niveles presentes en el videojuego, para tener la mayor comodidad al momento del desarrollo del videojuego se usaron diversas herramientas como Blender o Ninja Ripper 2 para la animación y modelados 3D. En el presente informe se detallarán los pasos realizados para la creación del videojuego Enigma del Tiempo, en el cual se explicará el propósito de cada herramienta implementada además de las metodologías implementadas.

**Abstract:** The video game Enigma del Tiempo is a project based on the Unity graphics engine, where graphic design and coding were utilized to create the various dialogues and levels present in the game. To ensure maximum comfort during the development process, various tools such as Blender and Ninja Ripper 2 were used for 3D animation and modeling. This report will detail the steps taken to create the video game "Enigma del Tiempo," explaining the purpose of each implemented tool as well as the methodologies used.

**Palabras clave:** enigma-del-tiempo, juego, unity, desarrollo de videojuegos, c#, stable diffusion, invoke ai, pixlr, 3d, realidad aumentada.

1. **Introducción.**

El diseño de los videojuegos durante años ha sido considerado como un medio de entretenimiento ideal para las personas que buscan una cómoda historia por resolver. El videojuego Enigma del Tiempo fue desarrollado por medio de Unity ubicado en el género de puzles/deportes, en el presente informe se redactará los pasos realizados para realizar el juego además de explicar sus herramientas de desarrollo.

1. **Desarrollo**

Según el experto en desarrollo de videojuegos [1], Unity es una plataforma de desarrollo de videojuegos la cual es muy utilizada en cuestiones de desarrollo de juegos y aplicaciones, en donde también se incluye tecnologías como la Realidad Virtual, la Realidad Aumentada, miniseries animadas, entre otras cuestiones. Dentro de sus características, se puede destacar que con Unity se puede trabajar en diversos capos, tales como el desarrollo de videojuegos, la realidad aumentada y virtual, producción animada, inteligencia artificial, multijugador e interfaz de usuario, en donde Unity destaca en cada característico debido al amplio marco de herramientas que ofrece para el desarrollo.

Además, el experto menciona diversos aspectos a tomar en cuenta a la hora de utilizar Unity, de los cuales pueden favorecer al desarrollador al momento de trabajar. Primeramente, hay que destacar que Unity ofrece diversas herramientas para poder monetizar los juegos, haciendo que se pueda implementar las micro transacciones dentro de los juegos y otras medidas para generar ingresos. También, las construcciones en la nube son un factor esencial ya que se ahorra el tiempo y recursos locales, y Unity cuenta con esa característica. También, la colaboración es un punto muy importante debido a que Unity ofrece diversas herramientas para poder colaborar con otros desarrolladores dentro del mismo proyecto, similar a un control de versiones. Por último, estás las certificaciones Unity, las cuales validan el conocimiento y las habilidades del desarrollador.

También hay que tomar en cuenta la comunidad y soporte que ofrece Unity. Aparte de la documentación oficial de esta plataforma, la cual explica detalladamente ciertos factores y herramientas, también se puede participar en foros o plataformas de las cuales ofrecen un soporte no formal, pero a la vez al tener una comunidad unida hay una ayuda entre todos.

* 1. **Visual Studio Code**

Según [2], considera que Visual Studio Code es un editor de código fuente de código abierto, el cual tiene soporte para lenguajes de programación como Javascript, Typescript y el entorno de ejecución multiplataforma Node.js, también se puede utilizar otros lenguajes de programación por medio de extensiones que contiene este editor de código. Lo destacable de este editor es que viene con una terminal integrada, integración con Git, depuración de código y diferentes opciones para poder trabajar con contenedores o vms.

* 1. **.NET desktop development**

.NET desktop development es una agrupación de herramientas orientadas a la elaboración y desarrollo de aplicaciones para Windows creados por Microsoft. La información proporcionada por [3] indica que a través de Visual Studio mediante C# se puede implementar las utilidades que ofrece .NET, además de sus comodidades como funciones de edición del código, cómoda navegación y análisis eficaz del proyecto, estas características convierten a este conjunto en las preferidas por algunos desarrolladores de aplicaciones de escritorio.

* 1. **C#**

Es un lenguaje de programación creado por Microsoft el cual es parte del conjunto de herramientas .NET, este lenguaje fue creado principalmente como una variación moderna de la programación orientada a objetos, por lo que es usado en la mayoría de los proyectos como aplicaciones web, aplicaciones de escritorio, aplicaciones móviles y servicios en la nube. Según [4] este lenguaje es de código abierto y multiplaforma, además de que .NET como tal es realmente rápido y eficaz al momento de trabajar, pues requiere de menos potencia de procesamiento y ofrece buena flexibilidad y entendimiento para los desarrolladores.

* 1. **Blender**

Según la artista [5], Blender es un programa de software libre el cual sirve como herramienta para realizar modelados 3D, escultura, composiciones y animación. Además, este software es muy utilizado en el ámbito profesional debido a la amplia gama de opciones que trae para poder trabajar. Por otro lado, también puede servir como programa intermediario al momento de tratar de exportar un archivo a otro formato, ya sea para utilizarlo en software de desarrollo de videojuegos, tales como unity o unreal engine, o utilizarlo en otros programas.

Entre sus funcionalidades principales, se puede destacar la creación de modelos tridimensionales, escenarios, objetos u elementos. También ofrece diversas herramientas las cuales favorecen en la animación de personajes y objetos, en donde también se incluye la capacidad de poder implementar plugins o huesos. En cuestiones de impresión 3D, facilita el proceso y la exportación en lo que es formato obj y stl.

* 1. **Ninja Ripper 2**

Según los desarrolladores de este software [6], es un software que destaca en la extracción de geometría de juegos 3d, para posteriormente exportarlo en editores, como Blender o Maya, en el cual puede ser de gran utilidad al momento de realizar mods, plugins o algún tipo de expansión de comunidad en los juegos.

* 1. **Invoke AI y Stable Diffusion**

Según [7], Invoke AI es una herramienta de código abierto con la capacidad de generar imágenes en la cual el usuario envía el texto y la imagen se genera, es decir, text to image. Además, ofrece diversas funcionalidades y opciones personalizables las cuales son herramientas factibles a la hora de trabajar con esta inteligencia artificial. Cabe destacar de que cuenta con una interfaz de usuario que se adapta a las necesidades de los usuarios, permitiendo la integración de herramientas como Illustrator o Photoshop.

Por otro lado, [8] menciona que Stable Diffusion es un motor de inteligencia artificial open source que principalmente fue diseñada para generar imágenes con la metodología text to image. También tiene la capacidad de generar imágenes con la metodología image to image, editar, añadir o borrar imágenes o elementos.

En conclusión, Stable Diffusion puede ser utilizado dentro de Invoke AI, siendo Invoke AI la interfaz gráfica y Stable Diffusion funcionando con toda la lógica para generar las imágenes.

1. **Materiales y Métodos**

Para poder desarrollar el juego, se emplearon diversas herramientas y técnicas las cuales fueron favorables en la creación del juego, haciendo que el juego tenga una interfaz adaptable al usuario y también un entorno interactivo y atractivo.

**Materiales:**

* Blender (Herramienta para poder exportar en un formato específico los modelos extraídos con ninja ripper)
* Stable Diffusion (Inteligencia artificial para generar imágenes)
* Pixlr (Editor de fotos para poder diseñar los diálogos)
* Invoke AI (Como interfaz gráfica de Stable Diffusion)
* Ninja Ripper 2 (Herramienta para extraer modelos de juegos)
* Unity (Entorno de desarrollo del juego)
* C# (Como lenguaje de programación)
* Visual Studio Code (Como editor de código)

**Métodos:**

Antes de empezar a crear el juego en el entorno de desarrollo Unity, se realizó una lluvia de ideas de cuál será la historia del juego. Luego, se empezó con la búsqueda de modelos para poder importar en Unity, como también se empezó hacer uso de la herramienta Ninja Ripper como extractor de modelos de juegos (tales como Garry’s mod y Roblox), como también se hizo un git clone del repositorio pb\_stl y su respectiva configuración. Pb\_stl es una herramienta esencial para poder importar los archivos con extensión stl.

Luego de haber reunido todos los modelos necesarios para el juego, se procedió con la ubicación y el ajuste de dimensiones de dichos modelos, para posteriormente configurar las colisiones de los objetos, como el panel de baloncesto, el suelo y la pelota.

Con respecto a la codificación de la pelota, se desarrolló un script con el lenguaje C#, ya que este es el lenguaje nativo de Unity. Se creó la lógica de la pelota con la funcionalidad de detectar el deslizamiento del dedo en la pantalla, calculando la dirección y fuerza del lanzamiento, como también se ajustó parámetros de velocidad y ángulo para poder lograr un comportamiento más natural y fluido de la pelota.

Por último, se diseñó un botón con label Repetir Tiro, el cual tiene la funcionalidad de que después de tirar la pelota, el usuario deba presionar ese botón y la pelota volverá a su posición de inicio.

Con respecto a la interfaz de inicio del juego, se diseñó una interfaz sencilla para más facilidad al usuario poder jugar el juego, con botones de salir y jugar, en donde el botón de jugar sirve para poder iniciar el juego y su historia, mientras que el botón salir pues simplemente abandona el juego.

Para la creación de los escenarios, se utilizó Invoke AI como interfaz de usuario de Stable Diffusion y Pixlr como editor de fotos. Primero, se generó diversas imágenes como los escenarios principales para los diálogos, y la forma en la que fueron generadas estas imágenes fue con la metodología text to image, el cual el usuario escribe un texto y la inteligencia artificial generará la imagen dependiendo del contexto del texto. Cabe recalcar de que se utilizó prompts específicos para obtener mejores resultados.  
En adición, elementos muy importantes que favorecieron para poder generar las imágenes fueron algunos modelos y conceptos, en donde mayormente se utilizó el modelo cartoonxl\_v10 y el concepto anime\_game\_icon, las cuales permitieron generar imágenes animadas. Dichos modelos anteriormente fueron sacados de la página civitai, que es una página la cual contiene muchos modelos que favorecen la generación de cualquier imagen de diversos estilos, sean 2d, 3d, animados, entre otros.  
Por otro lado, también se utilizó Stable Diffusion para la creación de los personajes de la historia, utilizando también prompts específicos, y esta vez se utilizó la opción de prompts ignore o prompts que no queremos que se tomen en cuenta, como lo es la mal formación de cara, brazos o alguna parte del cuerpo del personaje. Igualmente, aquí también se utilizó los modelos cartoonxl\_v10 y el concepto anime\_game\_icon para poder generar personajes animados.

Pixlr facilitó en la edición de las fotos para la creación de los diálogos, para posteriormente implementarlos dentro del juego como escenas seguidas en donde el usuario puede saltar dichos diálogos. Además, para el logo del juego se utilizó un ícono creado con Stable Diffusion y se realizó la respectiva edición para posteriormente importarlo en el menú principal del juego.

Para la creación del segundo nivel del juego se realizó un escenario de fútbol con sus debidos elementos tales como arco, pelota y gallina, los cuales se configuraron con sus respectivas colisiones y hitbox. La dificultad del juego se basó en la obstrucción del paso del balón hacia el arco a partir de una gallina que se mueve de lado a lado tal como si fuera un arquero, para lograr esta lógica de juego se añadieron las debidas codificaciones tanto para la pelota como para la gallina, las cuales permiten el desenvolvimiento del nivel otorgando cierto grado de dificultad para el jugador para poder marcar gol e ir sumando puntos para pasar el nivel. La lógica del nivel se basa en poder tirar el balón en dirección al arco sin que la gallina bloqueé el balón y para poder concluir el nivel se añadió el método de anotar, el cual obliga al jugador anotar tres veces para poder pasar al siguiente desafío.

Para la creación del tercer nivel, se buscó un modelo adecuado para el escenario de golf y se ajustó a nuestras preferencias. Esto incluyó modificar detalles visuales y estructurales del campo para que se adaptara al estilo del juego.

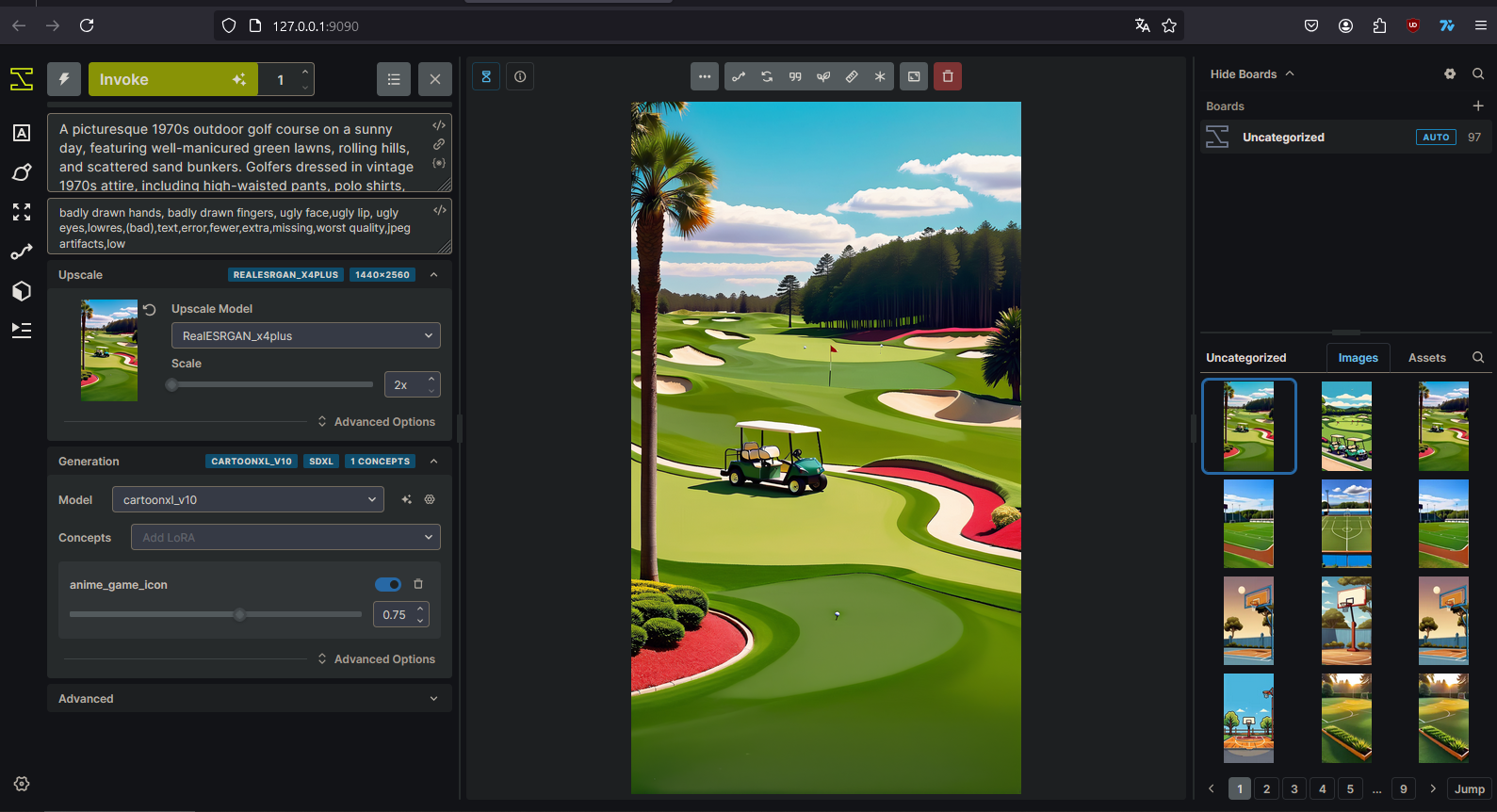
Luego, se configuraron las colisiones de los objetos en el campo de golf. Esto incluyó ajustar las propiedades físicas del hoyo, de manera que la pelota interactúe correctamente con él. Además, se calibró el hoyo para asegurarse de que la pelota pueda entrar y ser detectada adecuadamente.

Finalmente, se implementó la lógica y la codificación necesarias para que el juego reconozca cuando la pelota entra en el hoyo. Esto implica programar una condición que verifica si la posición de la pelota coincide con la del hoyo. Si la pelota se encuentra dentro del hoyo, se registra como un punto en el juego. Esta lógica asegura que el sistema de puntuación funcione correctamente y ofrezca una experiencia de juego fluida y precisa.

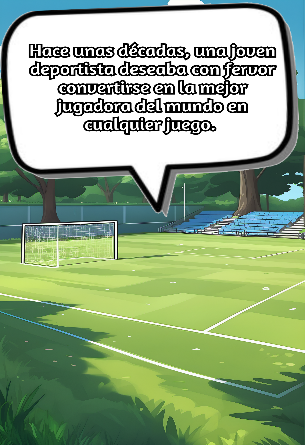
Posteriormente, para la creación de los diálogos se diseñaron escenas nuevas en las cuales se proyectan la continuación de la historia y la información sobre el desafío a cumplir, además de esto se agregó un botón para pasar de escena en escena del diálogo.

1. **Resultados y Discusión.**

La siguiente imagen es uno de múltiples resultados que se obtuvo al momento de generar las imágenes, especificando lo que se desea por medio de prompts.

****

Mencionado anteriormente en la metodología, Pixlr fue la herramienta que facilitó la creación de las escenas y diálogos. Estos fueron algunos de los resultados que se obtuvo al momento de realizar la edición de las imágenes para posteriormente importarlas al juego.

Resultado final de la generación de imagen del menú y logo del juego.

Resultado final del menú del juego dentro de la aplicación, en el cual se le agregó botones de jugar y salir.



Uno de los diálogos principales de la historia dentro de la aplicación.

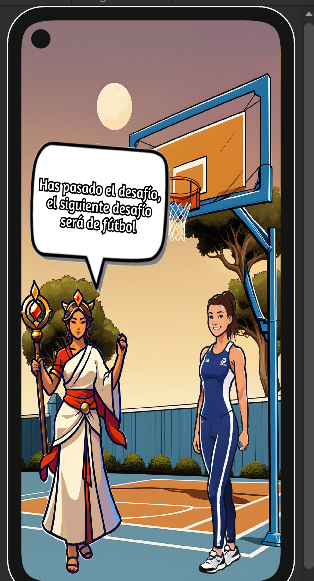


Primer desafío de la historia, el cual consiste en que el jugador realice cierta cantidad de canastas hasta alcanzar la puntuación requerida para pasar al siguiente desafío y también seguir con la historia.

Imagen que contiene Aplicación

Descripción generada automáticamente

Uno de los diálogos para el siguiente desafío.



El segundo desafío, el cual consiste en realizar cierta cantidad de goles para poder pasar al último desafío, como también seguir con la historia del juego.

Una pantalla de televisión

Descripción generada automáticamente

Diálogo de la transición del campo de fútbol hacia el campo de golf, para el último desafío.



Último desafío, el cual consiste en realizar cierta cantidad de hoyos, para finalmente cumplir con el objetivo de finalizar todos los desafíos, como también finalizar con la historia del juego.

Imagen que contiene monitor, reloj, pantalla

Descripción generada automáticamente

Diálogos finales, en donde después de haber finalizado todos los desafíos, la deidad devuelve al presente a la protagonista y finaliza la historia.

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza media

Los resultados previamente presentados cumplen con los requisitos establecidos para una experiencia de juego envolvente e interactiva; por lo tanto, son los esperados. Los diálogos fomentan la narrativa de manera efectiva, la interfaz de usuario es intuitiva y atractiva, y cada desafío tiene una mecánica clara y motivadora. Los jugadores pasan por una serie de pruebas que conducen a una conclusión satisfactoria, donde la protagonista, habiendo aprendido valiosas lecciones, regresa a su tiempo.

Los desafíos enfrentados durante el desarrollo del videojuego fueron básicamente por la falta de conocimiento general de cómo realizar un videojuego, sin embargo, se pudo superarlos gracias a tutoriales y guías gratis encontradas vía internet, de esta manera de pudo lograr crear los diálogos y niveles del videojuego y por ende concluirlo.

1. **Conclusiones.**

Las conclusiones obtenidas se basan en los resultados del presente proyecto en los cuales se determinó que el progreso del videojuego desde su inicio fue progresando nivel por nivel estableciendo los determinados métodos de jugabilidad y dificultad, además de que se consideraron los requerimientos y características para que las diversas herramientas utilizadas puedan ejecutarse con el objetivo de lograr diseñar y crear los niveles.

Se consideró también ideas de videojuegos creados como guía de preparación y desarrollo de Enigma del Tiempo y una vez recopilado toda la información se pudo realizar la historia y la jugabilidad del videojuego. Cabe recalcar que es importante tener materiales ideales para la creación de este tipo de proyectos ya que al usar procesamiento gráfico tiende a explotar el rendimiento total de los componentes.

**Referencias Bibliográficas.**

|  |  |
| --- | --- |
| [1] | D. Erosa García, «OpenWebinars,» 27 07 2024. [En línea]. Available: https://openwebinars.net/blog/que-es-unity/. [Último acceso: 27 07 2024]. |
| [2] | F. Flores, «openwebinars,» 22 07 2022. [En línea]. Available: https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/. [Último acceso: 22 07 2024]. |
| [3] | Microsoft, «Visual Studio IDE con .NET: desarrolle cualquier aplicación con C#, F#, VB,» 15 mayo 2023. [En línea]. Available: https://visualstudio.microsoft.com/es/vs/features/net-development/. [Último acceso: 31 julio 2024]. |
| [4] | Microsoft, «C# El lenguaje de programación moderno, innovador y de código abierto para crear todas sus aplicaciones.,» [En línea]. Available: https://dotnet.microsoft.com/es-es/languages/csharp. [Último acceso: 31 julio 2024]. |
| [5] | A. Chirivella González, «profesional review,» 20 02 2022. [En línea]. Available: https://www.profesionalreview.com/2022/02/20/blender-que-es-y-para-que-se-utiliza/. [Último acceso: 31 07 2024]. |
| [6] | blackninja, «ninja ripper,» 2024. [En línea]. Available: https://ninjaripper.com/about. [Último acceso: 01 08 2024]. |
| [7] | V. Arnold, «neuroflash,» 27 11 2023. [En línea]. Available: https://neuroflash.com/es/blog/puede-invoke-ai-llevar-su-proceso-creativo-al-siguiente-nivel/. [Último acceso: 04 08 2024]. |
| [8] | Y. Fernández, «Xataca,» 30 01 2023. [En línea]. Available: https://www.xataka.com/basics/stable-diffusion-que-que-puedes-hacer-maneras-para-usarlo-tu-ordenador. [Último acceso: 04 08 2024]. |