

## Diseño de un videojuego aplicando una arquitectura en capas basada en el framework de unity

Aguas Luis <sup>1</sup>; Recalde Henry <sup>2</sup>

<sup>1</sup> Universidad Tecnológica Israel, Quito-Ecuador, [laguas@uisrael.edu.ec](mailto:laguas@uisrael.edu.ec)

<sup>2</sup> Universidad Tecnológica Israel, Quito-Ecuador, [hrecalde@uisrael.edu.ec](mailto:hrecalde@uisrael.edu.ec)

---

**Resumen:** El desarrollo de videojuegos es un tema especial en el campo del desarrollo de software. El equipo de desarrollo multidisciplinario y los componentes artísticos introducidos por este estudiante de ingeniería de software hicieron que los métodos de desarrollo tradicionales fueran ineficaces. Aquí, las ideas y mecanismos propuestos deben ser probados para entender si son efectivos y si contribuyen a las expectativas puestas en ellos. Esto significa que el proceso de desarrollo debe ser iterativo, porque si bien hay una hoja de ruta desde el principio y una descripción de lo que desea hacer, las ideas y los mecanismos del juego siguen apareciendo a lo largo del proceso de desarrollo, se modifican y algunos se revisan. La preparación técnica de expertos incluye el dominio de las últimas herramientas y avances de software. En la actualidad, ciertas plataformas se destacan y permiten la creación a partir de disciplinas de ingeniería de software y modelos de programación, entre ellas Unity, cuya practicidad es comparable a frameworks como Genexus y Visual Studio. Además, considerando que Unity incluye la compatibilidad de dispositivos y consolas del mercado.

**Palabras clave:** modelado, 3D, 2D, alas, vrml, renderizado.

### *Design of a video game applying a layered architecture based on the unity framework*

**Abstract:** Video game development is a special topic in the field of software development. The multidisciplinary development team and artistic components introduced by this software engineering student rendered traditional development methods ineffective. Here, the proposed ideas and mechanisms must be tested to understand if they are effective and if they contribute to the expectations placed on them. This means that the development process must be iterative, because while there is a roadmap at the outset and a description of what you want to do, game ideas and mechanisms keep popping up throughout the development process, are being modified, and some are being revised. Technical expert preparation includes mastering the latest tools and software advancements. Currently, certain platforms stand out and allow creation from software engineering disciplines and programming models, among them Unity, whose practicality can be compared to frameworks such as Genexus and Visual Studio. In addition, considering that Unity includes the compatibility of devices and consoles on the market.

**Keywords:** modelling, 3D, 2D, wings, vrml, rendering.

---

## 1. INTRODUCCIÓN

La industria de los videojuegos tiene un papel protagónico en la sociedad, se utilizan para la formación, la diversión o para este proyecto con fines educativos.

Con los cambios tecnológicos de hoy en día, los procesos y estrategias de aprendizaje han cambiado y lo que permite aprender a través de videojuegos, esto se logra mediante el uso de elementos interactivos que permiten a los usuarios aprender a través de logros y avances en el juego, y con el paso

del tiempo, el concepto de un Se forma la “Sociedad Digital” y gracias a la integración de los medios, ya no se asocia a ningún medio en particular, y los videojuegos se convierten en un objeto cultural y social que forma parte de nuestras vidas. diario.

Para nosotros, el contenido anterior es una de las motivaciones importantes para llevar a cabo el proyecto, ya que las nuevas herramientas existentes pueden mejorar la calidad del aprendizaje de la nueva generación, estableciendo así un nuevo modelo de enseñanza que de alguna manera potencie el aprendizaje de los estudiantes. aprendizaje significativo de los estudiantes. Considerando la situación anterior, se decidió proponer un escenario que permita analizar todas estas condiciones, en especial todos los contenidos relacionados con el campo técnico. Así mismo, como paso inicial y necesario, se diseñó y desarrolló un videojuego educativo que permitiera la enseñanza y dinámicas de enseñanza del principio de ingeniería de este tipo de análisis es lo que significa utilizar la formación de educación superior basada en las tecnologías de la información y la comunicación.

## 2. METODOLOGÍA

### 2.1 Metodología y arquitectura

La metodología permitirá definir las principales fases y roles que se encuentran inmersos en el desarrollo de un proyecto de software en este caso un videojuego, es necesario considerar ciertos aspectos que permitirán determinar el tipo de videojuego a desarrollar, tales como:

- Plataforma: PC, celulares, web, etc.
- Tipo: 2D o 3D.
- Para quién es: Niños, adolescentes o adultos.
- Motor de juegos: Unidad, Irreal, Cry Engine, etc.

Pero centrándonos en la metodología, es necesario establecer un mecanismo que nos permita ordenar el proceso de desarrollo de un proyecto de videojuego, en las siguientes fases (Metodología de desarrollo de proyectos con Unity en CIPSA, 2021).

### 2.2 Fases de la metodología

Para desarrollar un videojuego no existe un punto de partida predeterminado, sino que se deben considerar ciertas etapas para desarrollar el videojuego de manera ordenada. En cada etapa se resuelven diferentes problemas de producción de videojuegos, y en ella participan diferentes profesionales, las etapas se dividen de la siguiente manera:

#### 2.2.1 Fase de preproducción

Esta etapa tiene como objetivo definir los aspectos básicos de los videojuegos y dar un concepto de cómo se desarrollarán los videojuegos, lo que brindará modelos para la planificación y producción de videojuegos (Morales Urrutia, Nava López, Fernández Martínez, & Rey Corral, 2010). Además de los conceptos básicos de planificación, la idea también incluye proponer cómo ejecutar el proyecto y cómo organizar el proyecto según el equipo de trabajo, el presupuesto y el tiempo de ejecución. Si bien esta etapa contiene alrededor del 10% al 25%, es importante porque la idea o concepto a desarrollar aquí se configura de acuerdo a lo anterior y se crea el primer prototipo.

#### 2.2.2 Fase de diseño

En esta etapa se fusionan los conceptos definidos en la etapa anterior y se ubican en el documento de diseño del videojuego (game design document). Este documento propone todo el contenido que debe alcanzar un videojuego, incluyendo conceptos generales de alcance, público objetivo y propósito del juego, así como las diferentes dinámicas, mecanismos, roles, guiones y jugabilidad del video. Juegos para construir módulos, niveles y escenarios de videojuegos.

#### 2.2.3 Fase de planeamiento

Al igual que en el proyecto, en la fase de planificación se establecen y detallan diferentes actividades para cada campo o equipo de desarrollo de videojuegos. La idea en esta etapa es utilizar un método definido que permita el trabajo en equipo

para asignar todas las tareas a desarrollar, y que los videojuegos se puedan complementar y desarrollar de forma ágil (Acerenza et al., 2009).

#### **2.2.4 Fase de Producción**

En esta etapa, la codificación o programación de videojuegos, que se definió en la etapa anterior, es una de las más importantes y de mayor duración, ya que aquí el equipo de desarrollo de videojuegos realiza diferentes actividades, con el fin de dejar un Prototipo de juego. Básicamente, cada área de trabajo se lleva a cabo de acuerdo con el plan previamente completado. Estas áreas se pueden dividir en las siguientes partes: área de diseño de juegos, área de diseño de niveles, área de programación, área de diseño de personajes, área de diseño de ambientes, etc. Esta es la etapa donde entran todos los roles y todos los campos de conocimiento (Urrutia, 2010) que se resuelven con el desarrollo de videojuegos.

Todos los miembros del equipo de desarrollo iniciarán su tarea de acuerdo a las herramientas o software especificados en las fases de planificación y diseño. Con el tiempo, estas tareas serán realizadas por todos los miembros del equipo de acuerdo al cronograma y metodología definidos en la fase de planificación (Análisis de las herramientas Unity y Blender para el desarrollo de videojuegos con enfoque educativo, s/f).

En esta etapa, se puede dividir de diferentes maneras desde el concepto de diseño de personajes o el entorno del juego, un proceso que generalmente lo realiza un diseñador 3D o un artista del entorno. Después de eso, con personajes y entornos o mapas, comienza a desarrollar o programar todos los mecanismos y dinámicas que los jugadores podrán realizar en un videojuego para lograr un prototipo usable del juego. (Análisis de las herramientas Unity y Blender para el desarrollo de videojuegos con enfoque educativo, s/f).

#### **2.2.5 Fase de prueba**

Esta etapa solo incluye las pruebas del prototipo de videojuego realizadas en la etapa anterior. La idea es que estas pruebas corrijan todos los errores y mantengan el juego en la versión final para seguir vendiendo. El principal objetivo de esta etapa es

corregir todos los errores del prototipo, y en base a los comentarios de los testers o personas que han jugado la versión anterior del juego, hacer que el juego satisfaga las preferencias del público minimizando la cantidad de errores.

### **2.3 roles del equipo de trabajo**

El desarrollo de un videojuego es un proyecto el cual lo lleva a cabo un equipo multidisciplinario y por eso es importante definir cuáles son las funciones de cada miembro de este equipo, para lo cual se definirán los siguientes roles:

#### **2.3.1 Artista**

Los artistas suelen ser personas del ámbito del arte y el diseño gráfico, y son los encargados de crear todos los recursos gráficos de los videojuegos. Las responsabilidades principales incluyen crear y texturizar modelos de personajes usando animaciones correspondientes, elementos de escena definidos en el juego, niveles y varios contenidos 2D y 3D, así como crear imágenes, interfaces de usuario, diseños y promociones para usar como opciones en el juego.

#### **2.3.2 Diseñador**

El diseñador es quien define el proyecto y conceptualiza la idea a desarrollar, se involucran de principio a fin en todas las etapas del proyecto y en coordinación con todos los roles, una de las tareas más importantes del diseñador es transmitir al los miembros del equipo lo que se desea como producto final, definir las reglas mecánicas del juego, interacción con los usuarios, escenarios y niveles que tendrá el videojuego. Es decir, es el responsable del proyecto y el creador de la idea a desarrollar.

#### **2.3.3 Programador**

Su principal responsabilidad es crear el código del videojuego para plasmar todas las ideas definidas por el diseñador y utilizar los recursos creados por los artistas. Una de sus principales características es la creación de una arquitectura que permite la reutilización de código por parte de otros programadores que forman parte del equipo.

Los programadores son ingenieros informáticos con conocimientos de herramientas de desarrollo y frameworks especializados para proyectos de videojuegos. Una de sus funciones importantes es acordar con el proyectista todos los aspectos técnicos, riesgos y limitaciones que pueda tener el proyecto en base a su experiencia.

### 2.3.4 Probadores

El rol de testers y control de calidad se encarga de verificar si el juego está funcionando de acuerdo a los objetivos del proyecto. Su función principal es verificar que no haya errores o problemas de compatibilidad, y si hay suficientes errores de juego y acción. revisados.

## 2.4 Herramientas

El uso de herramientas es fundamental, sobre todo la adecuada selección de software para el desarrollo de videojuegos, para este proyecto se propone un framework como Unity y el lenguaje de programación C#.

### 2.4.1 Unidad

Unity es una herramienta para crear y desarrollar juegos digitales. Es un motor de renderizado de imágenes muy potente que se integra con otras herramientas de trabajo para ayudar a crear contenido 3D interactivo. Tiene innumerables recursos de alta calidad a través de la comunidad de usuarios, aquí también se puede intercambiar conocimiento, también es multiplataforma, por lo que puede crear juegos en iOS, Android, Web, Xbox 360, PS3, Wii, Windows Phone, Linux y más. otras plataformas (Unity Technologies, s/f).

Con las muchas herramientas, flujos de trabajo y capacidades de iteración rápida de Unity, puede crear juegos, simulaciones o visualizaciones en 2D y 3D. El motor de Unity te permite crear innumerables personajes, juegos, diseños, entornos, etc.; desde títulos de consola 3D hasta aplicaciones móviles 2D (Unity Technologies, s/f).

### 2.4.2 C#

Este lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft, permite la creación de aplicaciones sólidas de cualquier tipo, basando su programación en objetos y seguridad de

tipos, su sintaxis es fácil de aprender, lo que permite que cualquiera que conozca otro lenguaje de programación se familiarice rápidamente.

La programación es un elemento clave de cualquier plataforma de desarrollo de juegos, incluido Unity. El script que implementa el comportamiento de cada elemento del juego se puede implementar usando lenguaje C# y Visual Code o Visual Studio IDE. (Metodología de desarrollo de proyectos con Unity en CIPSA, 2021).

## 2.5 Arquitectura

En este caso, la arquitectura permite la construcción de un sistema, en este caso un videojuego, que se realizó en las primeras etapas de su desarrollo y permitirá su mantenimiento y crecimiento en el futuro. Por lo tanto, la arquitectura de software define la estructura que debe tener el producto de software, las partes que debemos desarrollar y la forma en que deben estar conectadas o relacionadas con el trabajo general. Se define en un alto nivel a través de una serie de patrones y abstracciones, que se utilizan en el desarrollo de software y la interacción entre sus diversas partes.

Según (García, 2014) Aunque existen muchos tipos y descripciones complejas de arquitectura de videojuegos, en su forma más simple, podemos considerar dos grandes piezas:

La interfaz es responsable de brindar retroalimentación a la experiencia del jugador procesando los comandos de entrada del usuario (teclado, mouse, etc.) y presentando el juego (a través de gráficos, motor de sonido, etc.)

El back-end se encarga de especificar las reglas de la lógica del juego, determinar la dinámica de interacción con los objetos, proporcionar comportamientos a los personajes no jugadores (NPC), coordinar sus interacciones, etc.

Se propuso la siguiente arquitectura para simplificar el funcionamiento de un videojuego en un bucle principal en el que se realizan tres acciones en cada iteración del bucle:

El motor del juego recopila la entrada del usuario y comunica los comandos que el jugador elige ejecutar a la vista lógica.

La lógica del juego determina lo que se debe presentar al usuario en función de los comandos de entrada del usuario, el estado actual del juego y las acciones realizadas por el NPC, y comunica esto al motor del juego. El motor del juego debe utilizar recursos gráficos y sonoros para transmitir el nuevo estado del juego al usuario.

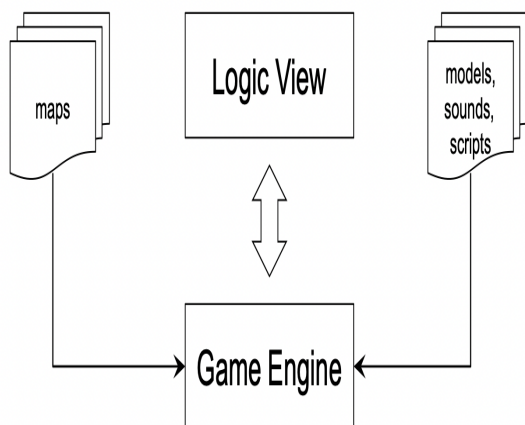


Figura 1. Simplearquitectura de videojuegos.

### 3. RESULTADOS Y DISCUSIÓN

#### 3.1 desarrollo de juegos

Para contextualizar el desarrollo de este videojuego, es necesario conocer el estilo que representará su implementación, es decir, qué tipo de videojuego se desarrollará, cuáles serán las limitaciones que esto conlleva y cómo es este tipo de videojuego. normalmente jugado (Marcos, 2012).

Para la aceptación pública de un videojuego se tienen en cuenta 3 factores que son totalmente relevantes a la hora de evaluarlo: jugabilidad, entretenimiento e historia; Es por ello que, por decisión del desarrollador, se realiza un videojuego tipo plataforma, que tiene como objetivo alcanzar una meta en un tiempo determinado, adquiriendo puntos, bonificaciones y evaluación en el terminal para cada escenario (Marcos, 2012).

#### 3.2 Servidor web

WebGL es una especificación estándar actualmente en desarrollo para mostrar gráficos 3D en un

navegador web. WebGL le permite mostrar gráficos 3D acelerados por hardware (GPU) en páginas web sin necesidad de instalar complementos en ninguna plataforma compatible con OpenGL 2.0 y OpenGL ES 2.0. Nos permitirá ejecutar los juegos que hicimos en Unity en el navegador sin instalar ningún complemento. Esta tecnología genera un script JavaScript, el cual convierte nuestro script C# a JavaScript para su posterior ejecución en la página web. Cuando construimos un proyecto WebGL, Unity crea una carpeta que contiene los siguientes archivos:

- Incruste el contenido en el archivo index.html de la página web.
- Un archivo JavaScript que contiene el código para el reproductor.
- El archivo. mem contiene una imagen binaria que se usa para inicializar el montón del reproductor. Un archivo .data que contiene datos de activos y escenas.

Algunos archivos JavaScript ayudan a inicializar y cargar el reproductor. Para reproducir el juego de forma estable, se inicia la página web que contiene toda la información del videojuego a partir de los archivos generados por Unity WebGL (Maths, s/f).

Para instalar el Manager y el Player en el servidor, es necesario importar las bibliotecas de Apache antes de generar el ejecutable del videojuego, para ahorrar tiempos de carga y contenido en segundo plano (teniendo en cuenta que el servidor de carrera local es Apache).

Finalmente se realiza un Launch Compilation que generará los archivos antes mencionados y una sencilla página web donde se almacenará el Unity Plug-in que permitirá jugar al videojuego.



Figura 2. Carga enchufable.

Como principales resultados obtenidos, luego de culminar el desarrollo del videojuego, se pueden mencionar los siguientes puntos:

- Un videojuego que admite componentes y funciones de nivel profesional, lo que permite a los usuarios finales sumergirse en una experiencia única.
- Jugabilidad, entretenimiento y narración, por eso se creó un videojuego basado en plataformas a decisión del desarrollador, con el objetivo de lograr un objetivo en un período de tiempo determinado, y obtener puntajes, bonificaciones y evaluaciones para cada escena en la terminal.

### 3.3 Caracterización de Personajes

Unity ofrece a los desarrolladores dos licencias: Unity gratuita y Unity Pro de pago. La versión gratuita tiene innumerables funciones que se pueden usar completamente para crear juegos. Unity Pro tiene una versión de prueba. Puede descargar la versión de prueba durante un mes y beneficiarse de las funciones adicionales proporcionadas por la versión Pro, incluido el analizador de memoria, la eliminación de oclusiones, los efectos de posprocesamiento y más. Para obtener la versión paga, debe comprar una licencia perpetua o suscribirse en la página de Unity (Unity Development Tool, s/f).

Para que el videojuego tenga una similitud entre la realidad y la ficción, se aplica una ligera similitud entre los personajes que interactúan en el videojuego, esto se basa en un diseño simple, que se refleja en la estructura del juego.

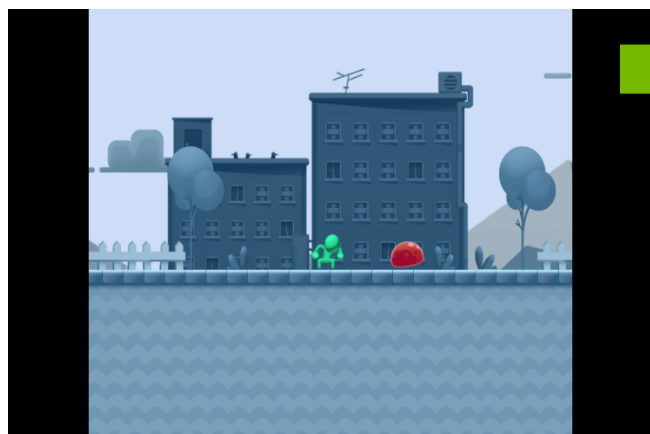


Figura 3. Juego de correr.

Otra característica a mencionar a la hora de desarrollar un videojuego son los elementos gráficos, encargados de dar vida a la programación del videojuego y lograr el gusto del usuario, se caracterizan por estar en un segundo plano en la aplicación con un objetivo puramente estético. (aunque en ocasiones interactúan con el protagonista del juego) y que aportan dinamismo a la hora de jugar (Unity Technologies, s/f).

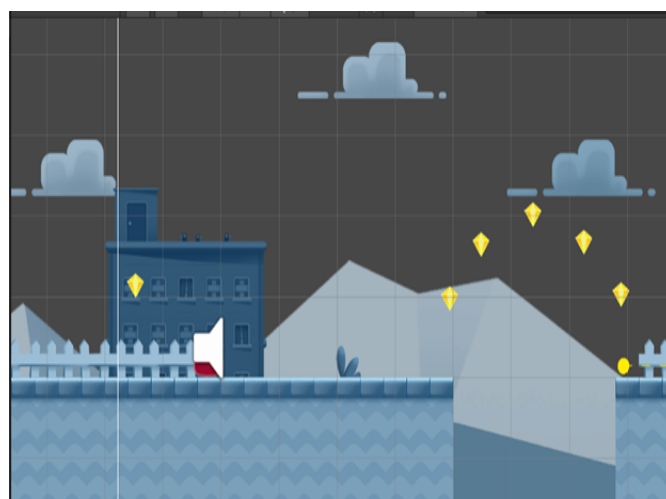


Figura 4. Estructura básica.

Otra de las características importantes en un juego es la incorporación de menús que consiguen una transición y un tiempo (tiempo de descanso) durante el cambio de escenario, así como un cierto control por parte del usuario en las configuraciones del juego, ya sea un control sobre música, configuración de botones, control de velocidad de personajes, etc, por eso se incorporó un menú de configuración

Una de las dificultades al inicio del desarrollo del videojuego fue lograr la creación de los niveles de una manera fácil, flexible y de bajo costo en memoria, lo que conlleva un gran problema, ya que un sistema de bajo costo y bajo costo Se requiere creador de escenarios. Aunque hay activos en el mercado que hacen un trabajo similar, no queríamos invertir en algo que no encajaba exactamente con lo que necesitábamos, así que por convicción decidimos crear un guión que nos permitiera crear los escenarios.

### 3.4 Creación de escenarios

Para evitar cargar todo el escenario a la vez en la memoria principal de un smartphone, se utilizó una antigua técnica en el desarrollo de videojuegos llamada tiling, que consiste en redibujar el escenario a través de instancias de pequeños fragmentos (en alusión al patrón de diseño Composite) (Candil, 2012).

Entonces, para la creación de escenarios en este proyecto se ha utilizado la estrategia antes mencionada en videojuegos antiguos, lo que asegura calidad en el escenario y bajo consumo de memoria principal, donde el escenario se genera a través de un archivo de texto que contiene el nivel dibujado con caracteres, que luego un Script intérprete es capaz de traducir y generar en la escena del juego, así la creación de niveles es más rápida, ya que solo requiere generar el archivo de texto ASCII que luego será traducido (Build completamente funcional et al., s/f).



Figura 5. Juego edificio.

Del mismo modo, convertir estos caracteres ASCII en widgets de Unity requiere muchos Prefabs (objetos prefabricados que se pueden instanciar sin

riesgo de ser modificados) que contengan una imagen renderizada del mismo tamaño; Así, lo que hace el generador es dibujar el Prefab, y luego posicionar un punto donde se generará el próximo Prefab, haciéndolo iterativamente hasta leer todo el escenario (Unity Technologies, s/f).



Figura 6. Ambiente.

### 3.5 Niveles de juego

El script para generar el escenario te permitirá definir los niveles más cómodamente; Es por esto que ahora se define el número de niveles a crear y lo que representa cada uno.

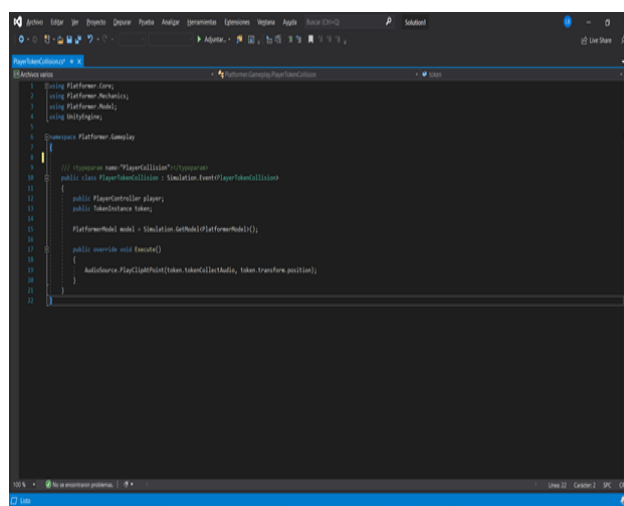


Figura 7. Ejecución Texto.

### 3.6 Paralelismo

Una de las técnicas más utilizadas últimamente en videojuegos y páginas web responsivas es el uso de parallax o parallaxing.



Parallax es una técnica de desarrollo 2D que logra dar un efecto de profundidad y proyección al escenario del videojuego a través del movimiento de los componentes más lejanos y más cercanos a la cámara, otorgando realismo y un acabado consistente, que alcanza una visualización más visual. agradable en el escenario.

La premisa del paralaje es el movimiento desigual de los componentes, debido a una visión subjetiva de la escena, esto significa que cuanto más lejos esté el objeto, su movimiento relativo será menor, a diferencia de los objetos más cercanos a la cámara o al observador, que se moverá más rápido frente al perceptor, desapareciendo del punto de vista mucho más rápido que los objetos distantes.

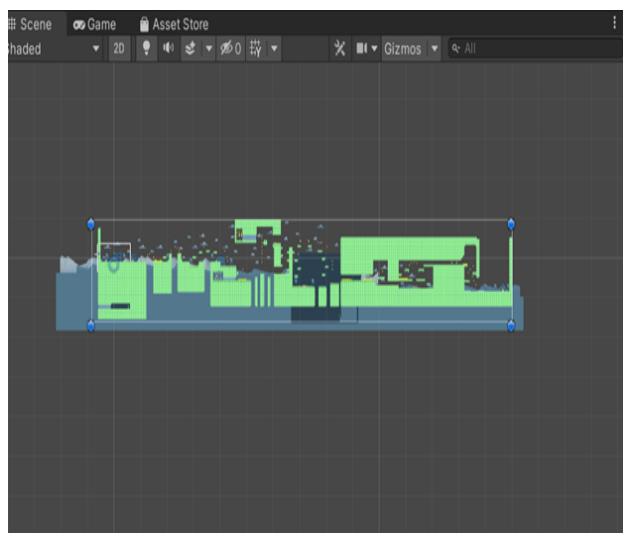


Figura 8. Paradajaaplicación.

### 3.7 Gestión de cámaras

En Unity y muchos otros motores gráficos libres, el uso de cámaras puede ser casi ilimitado (teniendo los componentes necesarios, mucha RAM y video) pero esto no hace que un videojuego con muchas cámaras sea de mejor calidad o más entretenido, ya que ese ahorro la memoria en un videojuego móvil es la prioridad por encima de todo. Teniendo esto en cuenta, es necesario administrar la cantidad de cámaras y el uso que se le dará a cada una, ya sea

para un uso similar o no, es totalmente ineficiente usar dos cámaras con el mismo objetivo.

El uso de múltiples cámaras se justifica solo si cada una tiene un uso único e irrepetible, cabe señalar que este es un componente que consume muchos recursos y abusando de su uso terminará consumiendo muchos recursos, como se sugiere en (Torque 3D, s / F).

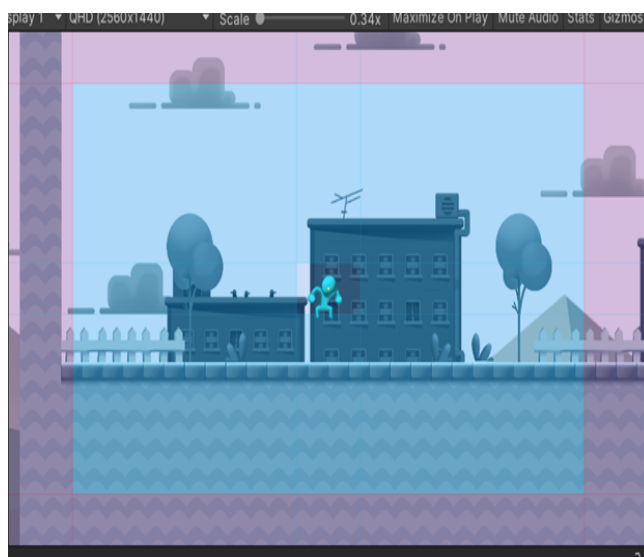


Figura 9. CámaraVista.

### 3.8 Inteligencia Artificial basada en Simple Reflex Agent

Dentro de la inteligencia artificial a implementar durante el proceso de desarrollo, se encuentra la de los enemigos del escenario, los cuales tienen un rango de visión del entorno ligeramente limitado y un comportamiento reactivo; Podría compararse con un simple agente reflejo, que junto con un poco de lógica difusa logra crear un personaje similar a un tercero que controla al enemigo. Este proyecto está orientado a conseguir hasta 4 personajes que se conviertan en enemigos escénicos, algunos con mayor nivel de dificultad que otros, cambiando su velocidad y reacción al interactuar con el enemigo.

Los enemigos son bolas rojas que actúan en función del movimiento del personaje.





Figura 10. Simple aplicación de estructura de agente.

### 3.9 Prueba de rendimiento

Estas pruebas tienen como objetivo determinar los requisitos mínimos para un uso óptimo del videojuego en diferentes dispositivos.

Para determinar el rendimiento en dispositivos móviles se utilizó una librería incorporada con Unity, la cual al ejecutarse con el dispositivo a evaluar vía USB muestra estadísticas sobre el funcionamiento del videojuego.

#### 3.9.1 Proceso

Se utilizó una resolución de proporciones 16:9, y se recorrió el juego de principio a fin, es decir, hasta derrotar al último enemigo. Para ello se utilizaron 3 dispositivos móviles, cada uno con capacidades diferentes.

Para evaluarlos uno contra el otro, se utilizó la herramienta Testing incorporada con Unity, que permite al Desarrollador visualizar ciertas características en modo depuración, para terminar aplicando una nota a la prueba.

**Tabla 1**  
 Especificaciones de características en Test en Dispositivos Móviles.

Característica	Prueba		
	Dispositivo 1	Dispositivo 2	Dispositivo 3
Fotogramas por segundo (FPS)	56 fps	45fps	25fps
Resolución proposicional	Sí	Sí	No
Marcha	7.1 de 10	6.5 de 10	4.3 de 10
VRAM utilizada	71 MB de 512 MB	71Mb de 256Mb	71Mb de 128Mb
RAM utilizada (promedio)	200Mb de 1Gb	200 MB a 512 MB	200 MB a 256 MB

Por tanto, al analizar los datos obtenidos tras realizar la prueba, se puede concluir que para ejecutar un videojuego sin problemas se requieren como requisitos mínimos las siguientes especificaciones:

- CPU: procesador Qualcomm Snapdragon 400 de doble núcleo a 1 GHz o similar.
- RAM: 512 MB para dispositivos de gama baja o media.
- Video: tarjeta aceleradora de gráficos con 256 MB o más de memoria de video dedicada.

Usando diferentes métodos de investigación, puede comprender completamente cada proceso y etapa necesarios para desarrollar un videojuego.

Además de aplicar todos los conocimientos aprendidos en el área de software para el desarrollo de videojuegos, se involucran diferentes campos relacionados con el diseño, la animación y el sonido, adquiriendo diferentes conocimientos básicos de cómo implementar y llevar a cabo los procesos involucrados para el desarrollo de lo que se propone.

A la hora de desarrollar un videojuego se debe pasar por todas las etapas y procesos necesarios para completar el proyecto, pasar por cada etapa de

desarrollo y adquirir los conocimientos necesarios para ejecutar cada etapa.

El desarrollo de videojuegos supone la fuerte participación de diversas disciplinas artísticas como el diseño gráfico, para lo cual debe afrontar nuevos retos, adquirir nuevos conocimientos y establecer intercambios con miembros de otras profesiones.

El uso de herramientas profesionales (como el motor Unity) ha ayudado mucho a la producción y desarrollo de software complejo (como los videojuegos). Así, por ejemplo, permite que los equipos de trabajo reduzcan costos, tiempos y complejidad a la hora de implementar proyectos. Por tanto, es muy útil encontrar tecnologías adecuadas que puedan aportar este tipo de ventajas a la hora de trabajar en un proyecto.

Al usar herramientas como Unity, la mayoría de las operaciones de la herramienta se basan en el uso de componentes y puede tener un software escalable que es fácil de mantener.

Además, cabe mencionar la importancia de elegir una herramienta de desarrollo con documentación y soporte completo, como Unity (frente a otras herramientas como Torque o GameMaker), que es un factor clave a la hora de utilizar y comprender todas las funcionalidades que Unity ofertas plataforma., no dejar de lado ayuda a resolver el problema tremendamente.

## REFERENCIAS

- Acerenza, N., Coppes, A., Mesa, G., Viera, A., Fernández, E., Lorenzo, T., & Vallespir, D. (2009). Una metodología para el desarrollo de videojuegos. Obtenido de <http://www.v.c.eh.en/jiwotvim/ISOFT2010-2011/Teoría/BlockIV/HTML.pdf>.
- Análisis de las herramientas Unity y Blender para el desarrollo de videojuegos con enfoque educativo. (s/f). Blogspot.com. Recuperado el 5 de noviembre de 2021, de <http://gmendezm.blogspot.com/2012/07/analisis-de-las-herramientas-unity-y.html>
- Cree entornos completamente funcionales, P. 3D GW y efectos dinámicos. (s/f). Aspectos básicos del desarrollo de juegos de Unity. Com.br. Recuperado el 29 de octubre de 2021, de <http://www.enucomp.com.br/2012/conteudos/minicursos/unity.pdf>
- Candil, D. (11 de octubre de 2012). Cuatro motores gráficos para perder el miedo y lanzarte al desarrollo de videojuegos. Vidaextra.com; Vida extra. <https://www.vidaextra.com/listas/4-motores-graficos-para-perder-el-miedo-y-lanzar-el-desarrollo-de-los-videojuegos>
- García, DL (2014). Metodología ontológica para el desarrollo de videojuegos. UNIVERSIDAD COMPLUTENSE DE MADRID.
- Herramienta de desarrollo de la unidad. (s/f). 1Biblioteca.co. Recuperado el 29 de octubre de 2021, de <https://1library.co/article/herramienta-de-desarrollo-unity-metodolog%C3%ADa-y-herramientas.myjdr22y>
- marcos (2012, 14 de noviembre). Motor gráfico... ¿y eso qué es? Wordpress.com. <https://marcosdiez.wordpress.com/2012/11/14/motor-grafico-y-eso-que-es/>
- Matemáticas, S. (s/f). transformaciones geométricas. Sangakoo.com. Recuperado el 29 de Octubre de 2021, de <https://www.sangakoo.com/es/temas/geometric-transformations>
- Metodología de desarrollo de proyectos con Unity en CIPSA. (2021, 22 de febrero). Cipsa.net. <https://cipsa.net/metodologia-desarrollo-proyectos-unidad-cipsa/>
- par 3D. (s/f). Garagegames.com. Recuperado el 29 de octubre de 2021, de <http://www.garagegames.com/products/torque-3d>
- Unidad de Tecnologías. (s/f). Cámara. Unity3d.com. Recuperado el 29 de octubre de 2021, de <https://docs.unity3d.com/es/current/Manual/class-Camera.html>
- Urrutia, GAM, López, CEN, Martínez, LFF y Corral, MAR (2015). Procesos de desarrollo para videojuegos. Cultura Científica y Tecnológica, (37).