

Creación de Videojuego 3D Numérico Bilingüe para dispositivos móviles

Creation of Numeric Bilingual 3D Videogame for mobile devices

Oscar Antonio Zárate Aguila^a, Alejandro Aguila Reyes^b,
Enrique González Tapia^c y César Ascencio Sánchez^d

Resumen / Abstract

Este proyecto tiene la finalidad de hacer uso del modelado tridimensional y la animación computarizada para crear una aplicación numérica bilingüe a manera de videojuego, de tal forma que sea una experiencia entretenida y productiva para el usuario. Al estar jugando, mediante la repetición de vocabulario, se logra la asociación de palabras en Inglés, logrando que la manera de aprender sea divertida. El videojuego está planeado para niños desde los 3 hasta los 10 años, pero puede ser empleado por personas de un rango de edad superior, cuenta con distintos niveles que brindan al usuario alternativas de juego con el fin de hacer el software más interesante y por consiguiente atractivo para el aprendizaje. Una ventaja adicional es la portabilidad, al ser una aplicación de Android para dispositivos móviles.

Palabras clave: software, videojuego, inglés, aplicación, Android, interactivo.

The main goal of this project is to use 3D modelling and computer animation to create a numerical bilingual videogame, so it can be a fun and productive experience for the user. By playing and repeating vocabulary, the user associates the words in English achieving the goal of learning by playing. The videogame is oriented to kids from 3 to 10 years old, but it can be used by people of higher range of age, it has different levels that give the user different game alternatives in order to make the game more interesting, hence, more attractive for learning. An additional advantage is the portability, since it is an Android App for mobile devices.

Keywords: software, videogame, English, application, Android, interactive.

^a. Profesor Investigador adscrito al Departamento de Ciencias Tecnológicas del Centro Universitario de la Ciénega de la Universidad de Guadalajara. Correo electrónico: oaza@cuci.udg.mx Orcid iD: <https://orcid.org/0009-0000-7678-7039>

^b. Profesor Investigador adscrito al Departamento de Ciencias Tecnológicas del Centro Universitario de la Ciénega de la Universidad de Guadalajara. Correo electrónico: aguilaalejandro@hotmail.com Orcid iD: <https://orcid.org/0009-0006-3932-5568>

^c. Profesor Investigador adscrito al Departamento de Ciencias Tecnológicas del Centro Universitario de la Ciénega de la Universidad de Guadalajara. Correo electrónico: enrique.gtapia@cuci.udg.mx Orcid iD: <https://orcid.org/0009-0009-7083-2842>

^d. Profesor Investigador adscrito al Departamento de Ciencias Tecnológicas del Centro Universitario de la Ciénega de la Universidad de Guadalajara. Correo electrónico: cascencio72@hotmail.com Orcid iD: <https://orcid.org/0009-0005-3994-1574>

1. Introducción

El proyecto V3DNB (Videojuego Tridimensional Numérico Bilingüe) tiene el objetivo de brindar una alternativa divertida en el aprendizaje del vocabulario básico del idioma Inglés y la realización de ejercicios numéricos a manera de videojuego.

Una de las formas más convenientes para lograr el aprendizaje, es de manera lúdica, por eso, se retoma este principio y se aplica al proyecto, en el cual los usuarios aprenderán sin darse cuenta, ya que lo verán como juego, no como tarea o trabajo.

Se emplea Blender para la creación y modelado de objetos 3D, a los cuales se les agregan animaciones, colores y texturas para después exportar estos elementos a Unity, donde se realiza la composición de escena, creación de interfaces, comportamiento de los elementos de juego e interacción con el usuario mediante codificación en el lenguaje de programación C#.

Una parte fundamental del proyecto es la integración de aspectos tecnológicos, creativos, lúdicos e intuitivos para facilitar el aprendizaje del usuario y se presenta como un videojuego portátil compatible con cualquier dispositivo móvil que cuente con sistema operativo Android.

Es cada vez más común que la tecnología esté presente en el día a día y principalmente en los niños quienes se interesan más por los dispositivos electrónicos, de hecho, es una de las razones por las que los niños se inclinan más a la utilización de nuevos métodos y técnicas acordes a la era tecnológica que les ha tocado vivir a las nuevas generaciones.

Al usar un software de modelado 3D, es posible crear un videojuego que incluya los puntos mencionados previamente además de que presentarán una ayuda para el aprendizaje del inglés.

Haciendo uso de las características de Blender, se puede realizar una gran variedad de acciones, como crear objetos, modificarlos, darles movimiento, cambiar color, manejar transparencias, reflexión, distintos tipos de iluminación, y al incorporarlo a Unity, se programa su comportamiento para que sea interactivo con el usuario.

El proyecto trata de unir aspectos creativos e intuitivos, así como tecnológicos para facilitar el aprendizaje del usuario. La interfaz creada trata de ser lo más amigable posible para que la interacción sea sencilla y llamativa.

2. Marco Teórico

2.1 Blender

Blender es un software dedicado principalmente al modelado, animación y creación de gráficos tridimensionales. Su principal atractivo es ser un software libre en su totalidad, compatible con todas las versiones de Windows, Mac OS X, Linux, Solaris, FreeBSD e IRIX, mientras que algunos otros softwares de diseño cobran una licencia por su uso, la cual puede alcanzar miles de dólares al año.

Blender se caracteriza por tener la disponibilidad de modificar la configuración original de acuerdo al gusto de cada usuario, lo cual le permite trabajar en un ambiente más acorde a las necesidades propias de cada diseñador, cuenta además con un motor de código interno, Python, lenguaje de programación de alto nivel que permite a los programadores, modificar comportamientos de los elementos creados.

Blender cuenta con las tres fases del diseño 3D, el modelado de objetos, la composición de escena y el procesamiento (rendering), que es la conversión del diseño en un ambiente tridimensional a una imagen o animación lista para ser exportada y visualizada fuera de Blender.

2.2 Unity

Unity es un motor de videojuegos que permite la publicación o compilación de videojuegos para múltiples plataformas comerciales.

Unity es un software gratuito que ofrece amplia gama de de destino de la publicación de juegos y es compatible con sistemas operativos de computadora como Windows y Mac OS X, así como para sistemas operativos móviles como IOS y Android además de navegación Web, PlayStation, XBOX, etc.

Unity permite la importación de objetos tridimensionales generados en otros programas, como Blender, por lo que se complementa a la perfección el uso de ambos softwares para la realización de este proyecto. Unity cuenta con un lenguaje de programación, C#, con las librerías correspondientes, propias de Unity, las cuales permiten la interacción en tiempo real y la compilación para la creación de aplicaciones independientes.

Las principales ventajas para emplear Unity en este proyecto son, la compatibilidad con Blender para importar elementos tridimensionales y la capacidad de crear aplicaciones móviles, las cuales son más fáciles de acceder para cualquier persona a diferencia de las otras plataformas donde es necesaria la inversión de un equipo más costoso, ya sea una consola (PlayStation o XBOX) o una computadora (Windows o Mac OS X).

Las razones por las cuales se utilizaron Blender y Unity para este proyecto se enumeran a continuación:

1. Ambos son gratuitos.
2. Son multiplataforma.
3. Blender es una herramienta muy completa de diseño y modelado.
4. Blender provee técnicas de modelado 3D que permite la creación de personajes llamativos.
5. Unity provee el código para la funcionalidad del juego y la interacción con el usuario.
6. Unity permite la generación de la aplicación para cualquier plataforma, en este caso Android.

3. Videojuego 3D numérico bilingüe para dispositivos móviles

Para crear el software V3DNB se utilizó Blender para modelar el personaje principal y elementos adicionales del juego, a partir de primitivas geométricas y herramientas de diseño, así

como la inclusión de esqueletos (Armatres)(Blender 2017) que facilitan el control de las diferentes animaciones involucradas.

Al terminar el proceso de modelado 3D en Blender, se exportaron dichos elementos de juego a Unity, motor de videojuego, en el cual se realizó la configuración de escenas, las interfaces gráficas de usuario para permitir la interacción en tiempo real, la programación en C#, soporte principal de la aplicación que determina el comportamiento correcto en cada momento, y por último, la generación de la aplicación móvil para dispositivos con sistema operativo Android.

Uno de los puntos más atractivos del proyecto V3DNB es la sensación de un ambiente tridimensional, es decir, perspectiva y profundidad, es mucho más llamativo que los sistemas convencionales en 2D donde sólo se aprecia verticalidad y horizontalidad.

La aplicación está desarrollada con una interfaz gráfica lo suficientemente amigable e intuitiva para que niños puedan emplearla, se añadieron, además, elementos llamativos y agradables a la vista para que la experiencia sea satisfactoria para el usuario.

Para la creación del software se siguieron estas etapas: Modelado de figuras, composición de escena y lógica de programación.

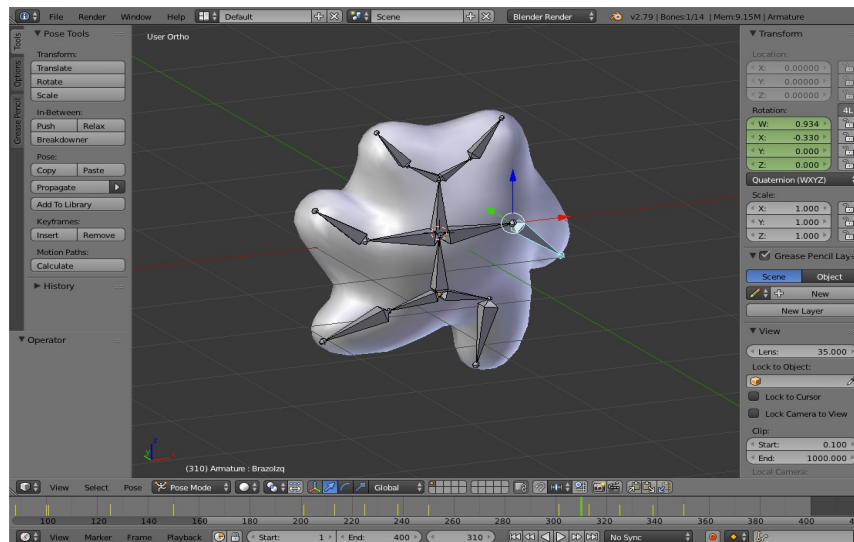
3.1 Modelado

La fase de modelado se refiere a la creación de cada uno de los elementos que intervendrán en la presentación final al usuario. Blender provee primitivas geométricas básicas (esferas, cubos, conos, cilindros, planos, etc.) (Primitives)(Blender 2017) las cuales se pueden transformar en cualquier elemento deseado para el diseñador mediante técnicas como la escultura (Sculpting)(Blender 2017) y la edición geométrica (Editing)(Blender 2017).

Para este proyecto, se modeló un personaje principal (Figura 1) a partir de un cubo, aplicando el modificador de multi-resolución para aumentar el número de caras y facilitar el proceso de diseño con la herramienta de escultura usando simetría en el eje X para modelar sólo un lado y dejar que Blender realizara el espejo del lado contrario para lograr la simetría en el personaje.

Después se le añadió un esqueleto (armature) que consiste, inicialmente, en un hueso, pero al acceder a modo edición, se generan huesos adicionales y se ajustan al modelo del personaje, una vez finalizado, se hace el proceso de emparentamiento para que el esqueleto controle la animación del personaje.

Figura 1
Creación y animación de personaje

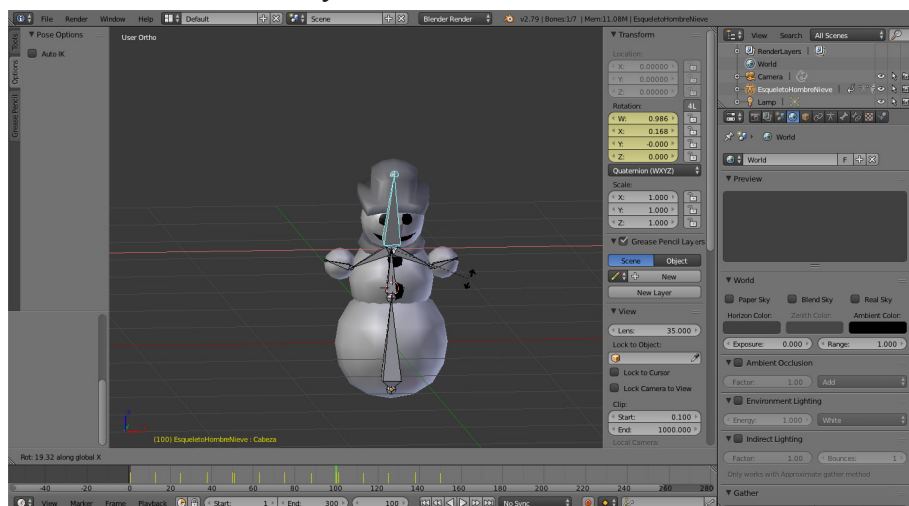


Elaboración propia.

La figura 2 muestra otro elemento de juego, un muñeco de nieve formado por múltiples figuras geométricas, esferas para el cuerpo, cabeza y manos, un cono para la nariz, un cilindro modificado geométricamente para el sombrero, ico esferas que representan los ojos y botones, curva de Bezier para la boca y un toroide para la bufanda.

Después, al igual que con el personaje principal, se añadió un esqueleto para para poder brindarle animación y hacerlo más llamativo en el videojuego.

Figura 2
Creación y animación de muñeco de nieve



Elaboración propia.

Considerando que la audiencia principal será el público infantil, se añadieron elementos adicionales que aparecen con forma de frutas y verduras, para hacerlo más llamativo. Dichos elementos fueron modelados a partir de esferas manipuladas, ya sea por herramientas de escultura (manzana) (Figura 3), deformación de enrejado (plátano) (Figura 4), extrusión de caras (aguacate) (Figura 5) o modificación de aristas y caras (calabaza) (Figura 6), por mencionar algunas técnicas, además se establecen los materiales que son los que llevarán los colores de cada uno de los elementos.

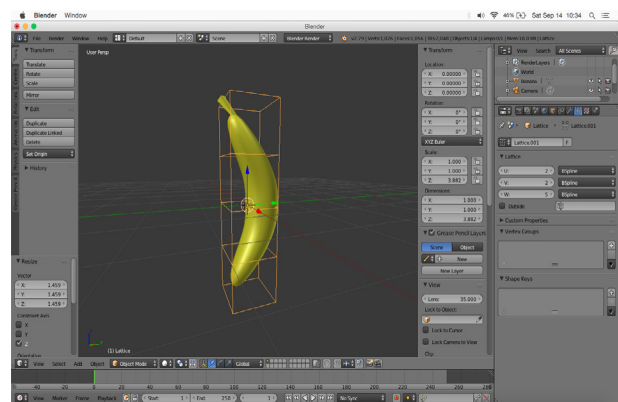
Estos modelos creados en Blender se exportan a Unity, donde se organizan los elementos de animación, y los materiales que llevarán para mostrarlos al usuario final

Figura 3
Creación de manzana



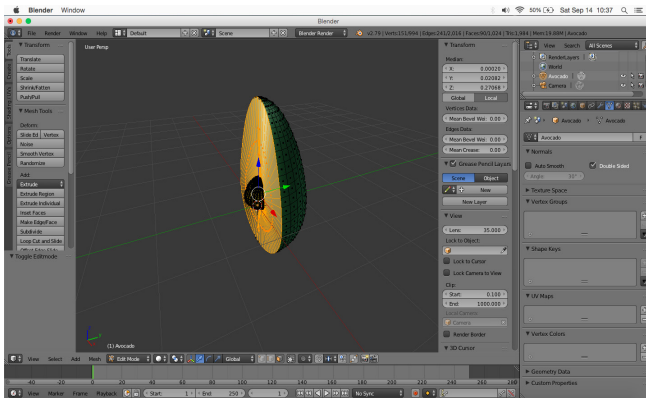
Fuente: Elaboración propia.

Figura 4
Creación de plátano



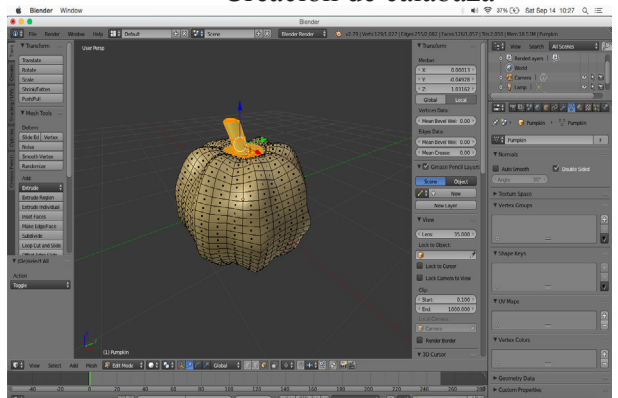
Fuente: Elaboración propia.

Figura 5
Creación de aguacate



Fuente: Elaboración propia.

Figura 6
Creación de calabaza



Fuente: Elaboración propia.

3.2 Composición de escena

Etapa en la cual se define el acomodo y la presentación de elementos al usuario, es decir, cómo se vera la aplicación final, es muy importante la correcta colocación de cámaras, iluminación y ángulos de visión que permitan al usuario disponer siempre de una perspectiva de juego amigable, de tal manera que sea llamativo y agradable para el jugador.

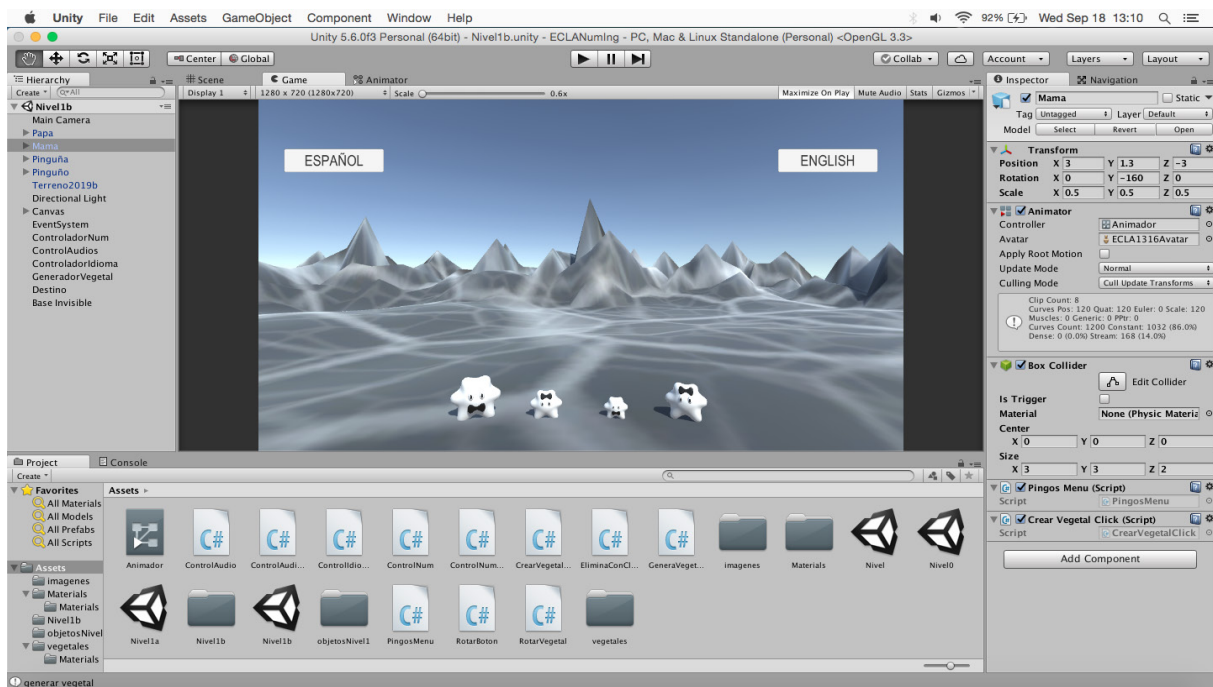
La composición de escena se realizó en Unity, así como la interfaz gráfica de usuario (GUI, Graphical User Interface) (Unity 2019) y el diseño del escenario combinando los elementos importados de Blender y los nativos de Unity.

Tomando en consideración el público a quien está dirigido el proyecto (niños) es necesario que la iluminación sea adecuada, para que sea fácil apreciar los detalles, mientras más colorido e iluminado el juego, más atractivo será.

Por contraste, se estableció un fondo rocoso (modelado en Blender) para una identificación fácil de los materiales (colores) usados para los demás elementos de juego. Se distribuyen los elementos, se define la interfaz correspondiente con los botones de interacción y por medio de codificación, se añade la animación y comportamiento de cada elemento de juego.

En la figura 7 se muestra la composición de escena en Unity, donde se define la posición de la cámara, el terreno y la iluminación correspondiente para los elementos de juego.

Figura 7
Composición de escena en Unity

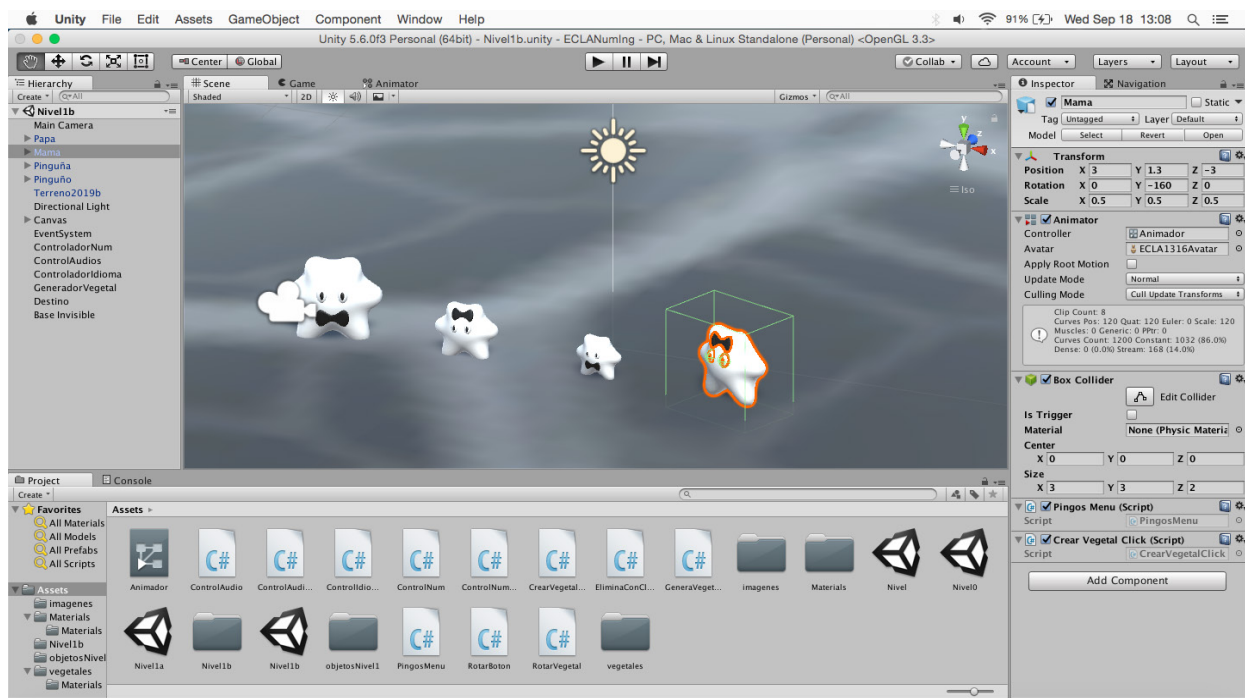


Fuente: Elaboración propia.

Una vez teniendo el terreno, cámara e iluminación definidos, se procede a la colocación de los elementos de juego, es decir, el personaje, los distintos modelos adicionales que se incluirán y se añaden las propiedades y componentes necesarios para su funcionamiento, como la adición de colisionadores, audios y el componente de código, cuya explicación se ampliará más adelante.

En la figura 8 se muestra la colocación de los personajes en la pantalla principal que se le presentará al usuario al iniciar el juego, donde se hará interacción con los personajes como darles distintas animaciones y cambiarlos de color.

Figura 8
Colocación de personajes



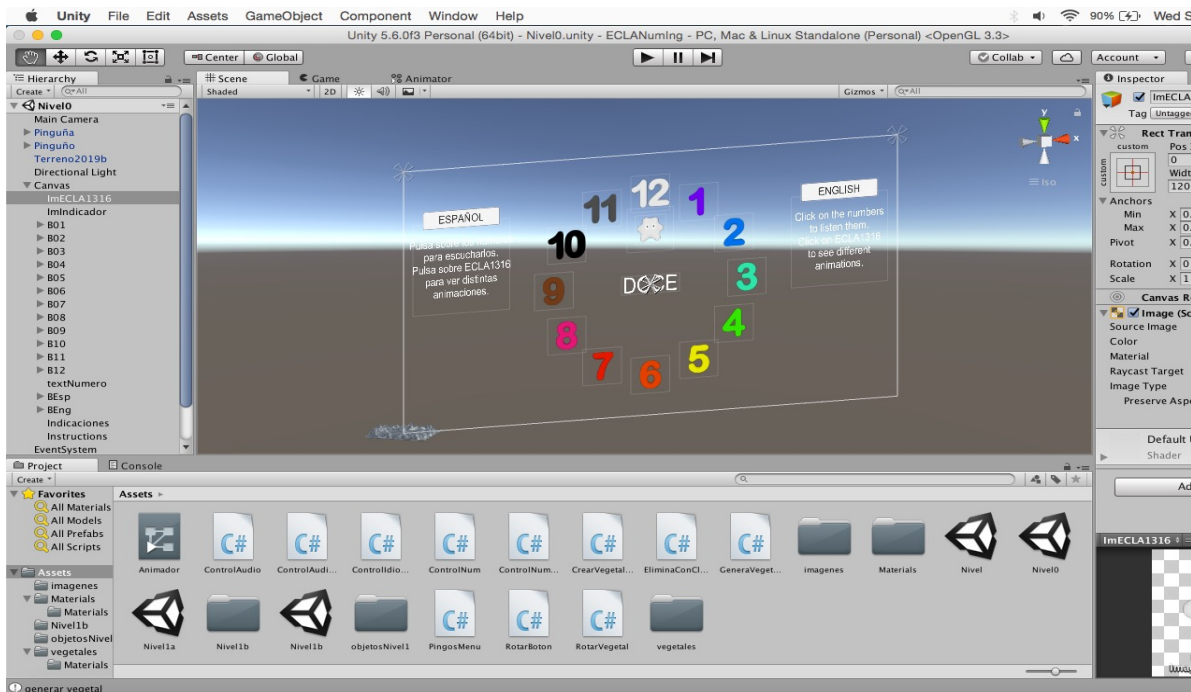
Fuente: Elaboración propia.

Un elemento indispensable en los videojuegos, o en cualquier software, es la interfaz gráfica de usuario (GUI, Graphical User Interface) ya que brinda interacción entre el usuario y la aplicación.

En la interfaz de usuario están los controles que el usuario puede utilizar, así como indicadores acerca del estado del videojuego como son contadores, barra de vida, colores a elegir, accesos a distintos modos de juego, etc.

En la figura 9 se muestra la creación y acomodo de una de las interfaces gráficas del videojuego, la pantalla de bienvenida al juego mostrando al usuario las dos opciones de idioma, Español e Inglés.

Figura 9
Creación de interfaz gráfica en la pantalla de bienvenida



Fuente: Elaboración propia.

3.3 Lógica de programación

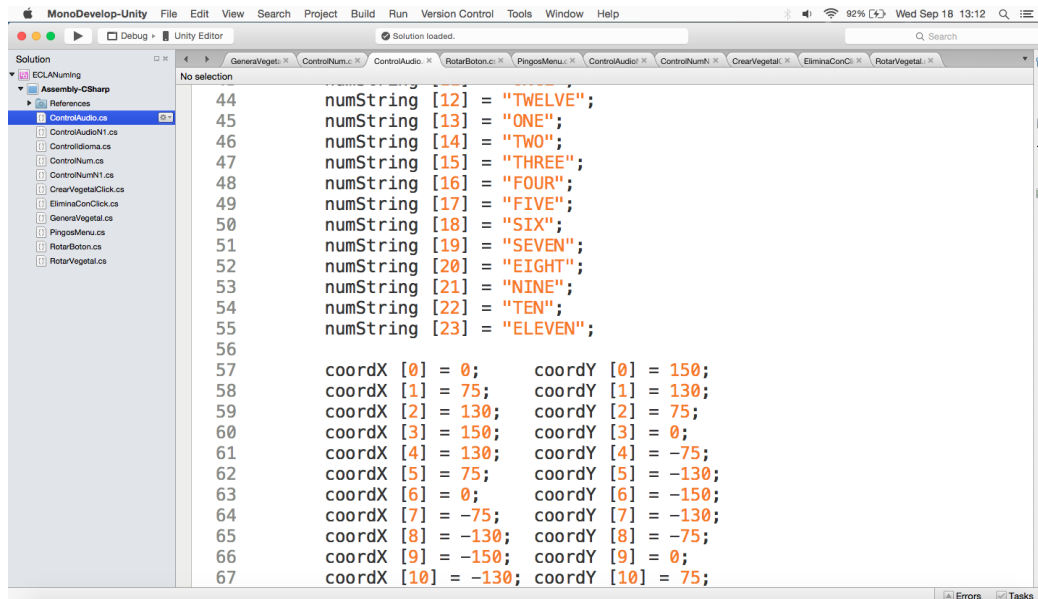
La etapa más importante de cualquier desarrollo de software es la que comprende tanto codificación como lógica de programación, es decir, los encargados de que los modelos diseñados previamente se comporten como videojuegos, además de que en esta etapa se establece el comportamiento de la interacción con el usuario, es decir, le permite al usuario y al programa establecer vínculo en tiempo real.

La programación o codificación se realizó en el lenguaje de programación C# en un módulo adicional de Unity llamado Mono Developer (Scripting)(Unity 2019), que es el ambiente en el que se programa el comportamiento de cada elemento del videojuego, así como verificar que no haya errores cuando el jugador este haciendo uso de la aplicación.

Para este proyecto, se emplearon más de 15 archivos de código, (clases), los cuales controlan los comportamientos del personaje principal, de los personajes auxiliares, los generadores de figuras, los cambios de color, así como los controles de interfaz y cambios de idioma.

La figura 10 muestra un segmento del código donde se muestra la elección del número y su distribución en pantalla con las coordenadas específicas.

Figura 10
Segmento de código de selección de número



```

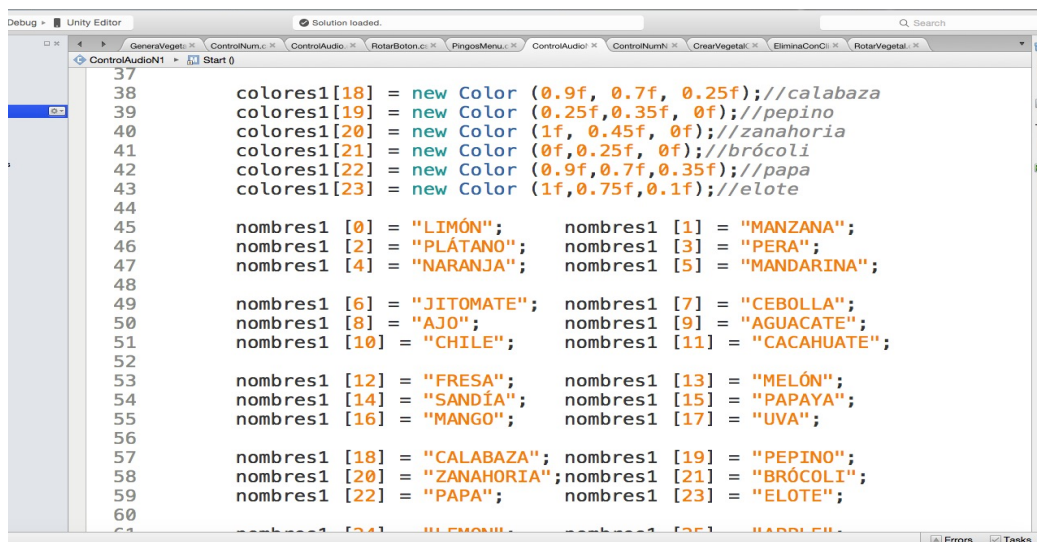
44 numString [12] = "TWELVE";
45 numString [13] = "ONE";
46 numString [14] = "TWO";
47 numString [15] = "THREE";
48 numString [16] = "FOUR";
49 numString [17] = "FIVE";
50 numString [18] = "SIX";
51 numString [19] = "SEVEN";
52 numString [20] = "EIGHT";
53 numString [21] = "NINE";
54 numString [22] = "TEN";
55 numString [23] = "ELEVEN";
56
57 coordX [0] = 0; coordY [0] = 150;
58 coordX [1] = 75; coordY [1] = 130;
59 coordX [2] = 130; coordY [2] = 75;
60 coordX [3] = 150; coordY [3] = 0;
61 coordX [4] = 130; coordY [4] = -75;
62 coordX [5] = 75; coordY [5] = -130;
63 coordX [6] = 0; coordY [6] = -150;
64 coordX [7] = -75; coordY [7] = -130;
65 coordX [8] = -130; coordY [8] = -75;
66 coordX [9] = -150; coordY [9] = 0;
67 coordX [10] = -130; coordY [10] = 75;

```

Fuente: Elaboración propia.

La figura 11 muestra un fragmento de código del comportamiento de los elementos de juego, generación de color e identificación de objetos.

Figura 11
Fragmento de código de cambio de color e identificación de objetos



```

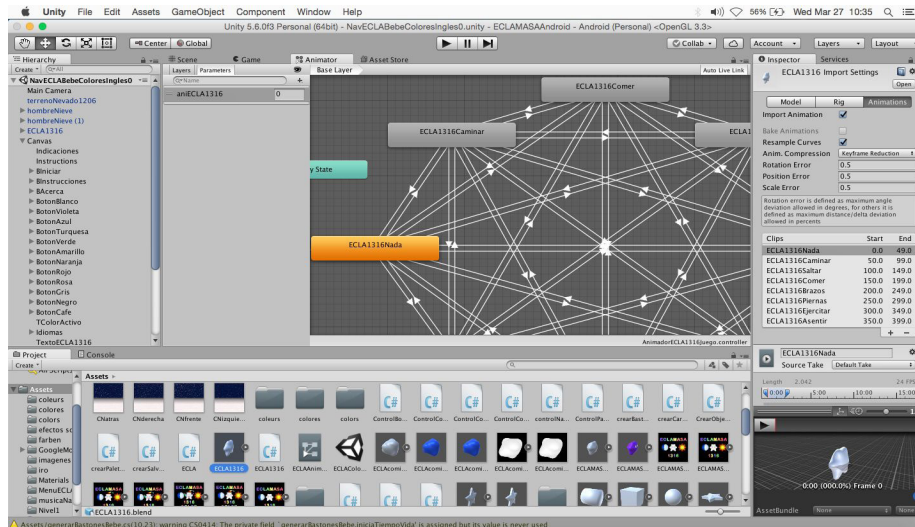
37
38 colores1[18] = new Color (0.9f, 0.7f, 0.25f); //calabaza
39 colores1[19] = new Color (0.25f, 0.35f, 0f); //pepino
40 colores1[20] = new Color (1f, 0.45f, 0f); //zanahoria
41 colores1[21] = new Color (0f, 0.25f, 0f); //brócoli
42 colores1[22] = new Color (0.9f, 0.7f, 0.35f); //papa
43 colores1[23] = new Color (1f, 0.75f, 0.1f); //elote
44
45 nombres1 [0] = "LIMÓN"; nombres1 [1] = "MANZANA";
46 nombres1 [2] = "PLÁTANO"; nombres1 [3] = "PERA";
47 nombres1 [4] = "NARANJA"; nombres1 [5] = "MANDARINA";
48
49 nombres1 [6] = "JITOMATE"; nombres1 [7] = "CEBOLLA";
50 nombres1 [8] = "AJO"; nombres1 [9] = "AGUACATE";
51 nombres1 [10] = "CHILE"; nombres1 [11] = "CACAHUATE";
52
53 nombres1 [12] = "FRESA"; nombres1 [13] = "MELÓN";
54 nombres1 [14] = "SANDÍA"; nombres1 [15] = "PAPAYA";
55 nombres1 [16] = "MANGO"; nombres1 [17] = "UVA";
56
57 nombres1 [18] = "CALABAZA"; nombres1 [19] = "PEPINO";
58 nombres1 [20] = "ZANAHORIA"; nombres1 [21] = "BRÓCOLI";
59 nombres1 [22] = "PAPA"; nombres1 [23] = "ELOTE";
60
61 nombres1 [24] = "LIMÓN"; nombres1 [25] = "MANDARINA";

```

Fuente: Elaboración propia.

Para poder incorporar correctamente en Unity las animaciones importadas de Blender, es necesario añadir controladores de animación (Figura 12), los cuales permiten la manipulación de las distintas animaciones del personaje, por medio de otro archivo de código.

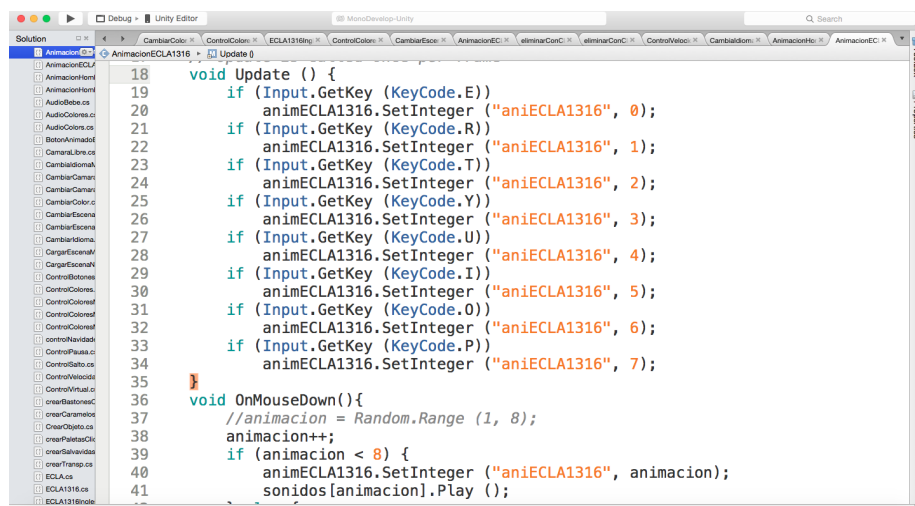
Figura 12
Controlador de animación.



Fuente: Elaboración propia.

La variable utilizada en el controlador de animación, se utiliza en los fragmentos de código donde se hace la interacción con el personaje para poder mostrar al usuario los distintos estados de animación (Figura 13).

Figura 13
Interacción entre código y controlador de animación

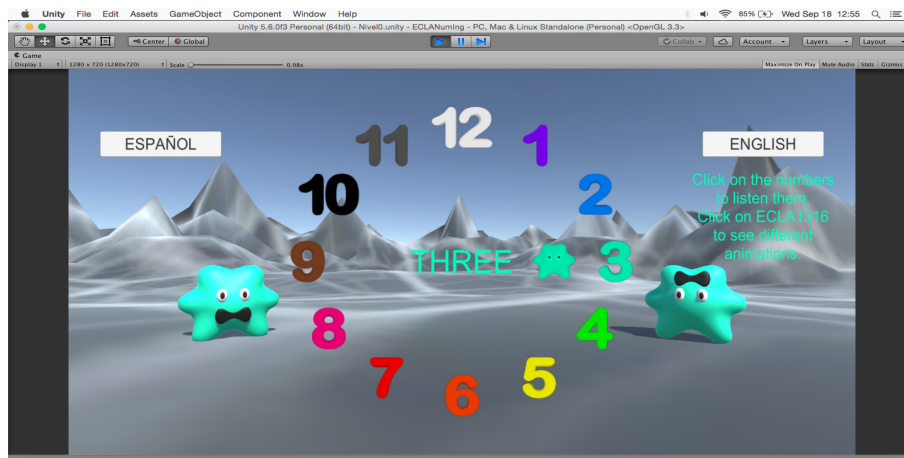


Fuente: Elaboración propia.

3.4 Interacción con el usuario

El proyecto V3DNB, tiene 3 modos de juego, el primero es la selección de números (Figuras 14 y 15), al pulsar sobre los personajes, se muestran las distintas animaciones, al presionar los números de la interfaz, se escucha el audio en el idioma activo y los cambios de color se muestran en los personajes.

Figura 14
Primer modo de juego (bienvenida) en inglés



Fuente: Elaboración propia.

Figura 15
Primer modo de juego (bienvenida) en español



Fuente: Elaboración propia.

En el segundo modo de juego (entretenimiento) (figuras 16 y 17), aparecen 4 personajes, al pulsar cada personaje, se generan 10 elementos aleatorios, de un total de 24 frutas y verduras, al presionar sobre uno de esos objetos generados, se cambia el color de los personajes y se reproduce el audio indicado, dependiendo del idioma elegido.

Figura 16
Segundo modo de juego (entretenimiento) en inglés



Fuente: Elaboración propia.

Figura 17
Segundo modo de juego (entretenimiento) en español



Fuente: Elaboración propia

En el tercer modo de juego (aventura) (figuras 18 y 19), aparece el personaje principal y hay elementos de juego distribuidos en el escenario, su tarea es recolectarlos, al inicio de cada aventura,

se cambia aleatoriamente el número de elementos a recolectar de cada tipo, al tocar cada uno de ellos, el personaje cambia al color del objeto recolectado, además de que se escucha el audio del color de acuerdo al idioma activo.

Otra curiosidad es que se muestran dos contadores, uno es el total a recolectar y el otro es el contador activo, el primero se disminuye y el segundo aumenta, logrando jugar de esta manera con sumas y restas de manera sencilla.

Al representar gráficamente los elementos requeridos, se muestra de manera simple la noción de valor posicional, ya que al ser seis elementos, se lograría contar hasta los cientos de miles para los usuarios que tengan mayor conocimiento numérico, sin embargo, no afecta la manera de juego para los más pequeños, porque lo ven de manera individual.

Figura 18
Tercer modo de juego (aventura) en inglés



Fuente: Elaboración propia.

Figura 19
Tercer modo de juego (aventura) en español



Fuente: Elaboración propia.

En el tercer modo de juego, se añadieron un control virtual para mover al personaje y un botón de salto para acceder a algunos objetos elevados, dando al jugador una experiencia similar a la de la mayoría de juegos de consolas.

4.- Conclusión

Gracias al avance tecnológico, es posible diseñar aplicaciones como ésta, las cuales son una alternativa lúdica al proceso de enseñanza, a medida que los software 3D han ido mejorando, se ha logrado el avance y desarrollo de muchos videojuegos más atractivos, por ende, que mejor manera de aprovechar esta tecnología que en los métodos de aprendizaje en los niveles básicos de educación. Una vez terminado el proyecto, se presentó dicho videojuego a grupos de usuarios distintos para que pudieran evaluar la aplicación V3DNB.

El videojuego ha sido empleado por individuos con edades distintas, desde pequeños de tres años hasta adultos de más de sesenta, siendo las opiniones, en general, satisfactorias, tanto en el diseño, como el funcionamiento y en los resultados obtenidos, siendo la intención de la aplicación, servir como auxiliar para el aprendizaje del idioma inglés. En etapas siguientes se planea añadir más elementos de juego y algún otro idioma.

La ventaja que ofrecen este tipo de aplicaciones es que proveen entretenimiento y aprendizaje, por lo que los niños (el público principal al que se destinó este proyecto) están ávidos de emplear esta tecnología.

En este proyecto se desarrolló una aplicación móvil empleando Blender y Unity con la intención de incentivar el aprendizaje numérico de manera lúdica, además de brindar la opción de jugar en Inglés, además de Español.

En este proyecto se creó un videojuego como aplicación móvil empleando Blender y Unity con la intención de mostrar una alternativa numérica bilingüe atractiva e interesante.

Se realizaron encuestas con preguntas clave para saber las opiniones del software y realizar mejoras de acuerdo a las sugerencias.

Las primeras etapas del proyecto son laboriosas porque comprenden la creación desde cero de todos los elementos requeridos para la integración de la aplicación.

Cada objeto fue creado en Blender a partir de las primitivas geométricas y herramientas de modificación hasta obtener los elementos deseados, empleando distintas técnicas como la edición de vértices, escultura o subdivisión de superficie, cabe mencionar que Blender no es un software intuitivo, por lo que es requerido tener experiencia para utilizar correctamente los modificadores necesarios.

Una vez terminados los diseños, se procedió a la exportación de los mismos a Unity, para crear escenarios, niveles de juego, interfaz gráfica y programación del funcionamiento de la aplicación, para ello, es necesario saber programación orientada a objetos, específicamente en C# con las bibliotecas correspondientes de UnityEngine que permiten al programador controlar la funcionalidad del software y su interacción con el usuario.

Siempre hay un respaldo de código para el comportamiento de cada objeto, por ende, sin conocimiento de programación, no es posible realizar videojuegos de este estilo.

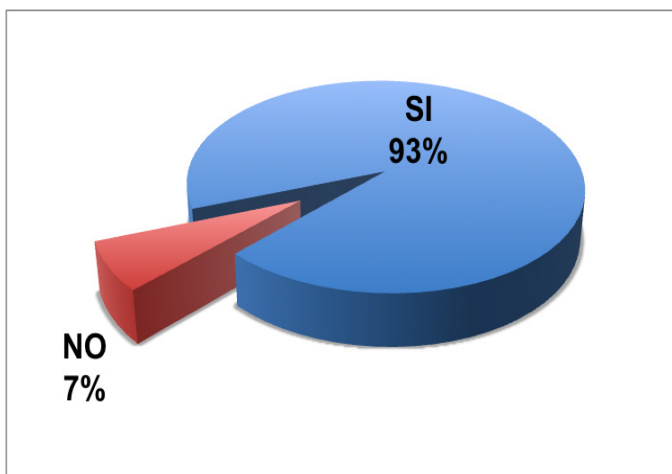
Una vez finalizadas las tres etapas de creación del videojuego (modelado, composición de escena y lógica de programación) sigue la encuesta de satisfacción.

En dicha encuesta, los usuarios expresan sus opiniones, al final, son ellos quienes determinan la aceptación del software o sugerencias de mejora, si hay necesidad de realizar modificaciones, de acuerdo a su particular punto de vista o del punto de vista de los padres del usuario, recordando que la audiencia es, en su mayoría, público infantil.

En cuanto a la simplicidad de la navegación en el videojuego y su utilización, hubo aceptación generalizada (figura 20), cumpliendo el objetivo principal de cualquier software, la sencillez de la interfaz, permitiendo que el usuario haga uso del software de manera amigable y que no sea una experiencia tortuosa por la complejidad de la interfaz.

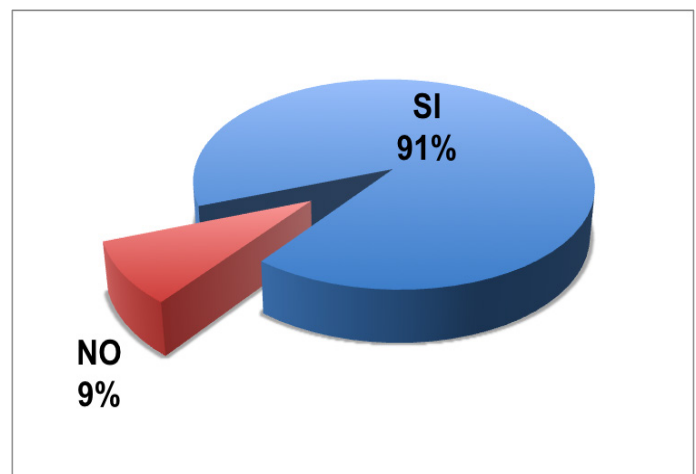
Otro requisito importante es que la aplicación resulte de interés para el usuario, como se aprecia la gráfica de la figura 21, al captar la atención del usuario, es más fácil cumplir el objetivo oculto, que el usuario practique tanto aritmética como inglés sin darse cuenta al estar entretenido en el videojuego.

Figura 20
¿Es la aplicación intuitiva?



Fuente: Elaboración propia.

Figura 21
¿Es atractiva al usuario?

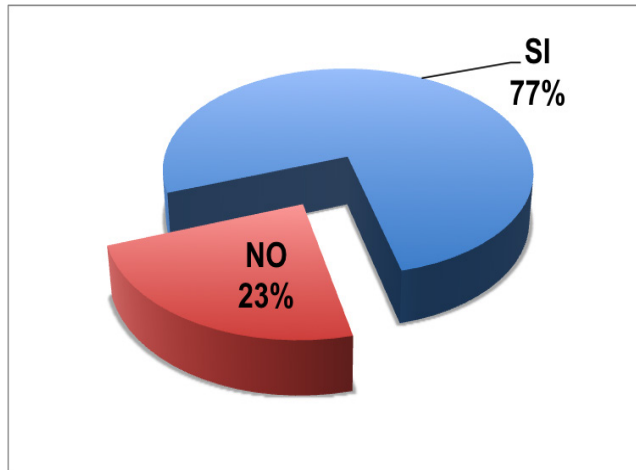


Fuente: Elaboración propia.

Después de emplear el videojuego, se realizó una prueba rápida de vocabulario a los participantes. Los niños pequeños, sobre todo, mostraron gran retención de vocabulario gracias a la repetición del audio durante el juego, es decir, aún sin darse cuenta, aprendieron palabras nuevas.

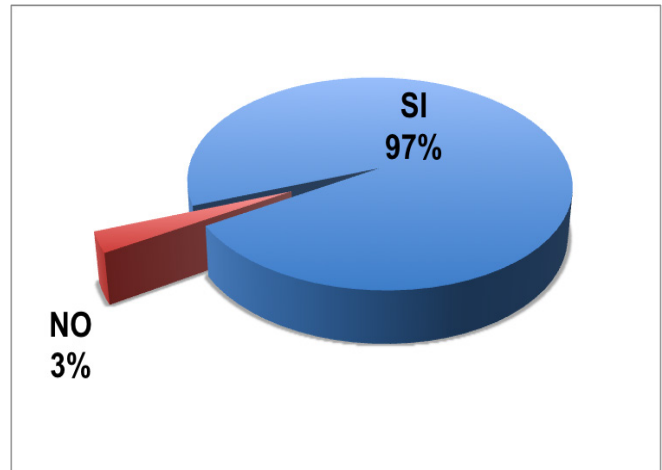
Los resultados se muestran en la figura 22. De acuerdo a la mayoría, el juego es llamativo y al preguntarles si les gustaría instalarlo o descargarlo, la gran mayoría aceptó, ya que les gustó el concepto de tridimensionalidad y funcionalidad en dispositivo móvil. (Figura 23).

Figura 22
¿Mejoría en el vocabulario?



Fuente: Elaboración propia.

Figura 23
¿Instalarían la aplicación?



Fuente: Elaboración propia.

Los resultados confirman que es viable la utilización de videojuegos como una alternativa didáctica, ya que a los menores (público destino) les llama la atención y aprenden sin darse cuenta, siendo el objetivo principal, lograr que el proceso de enseñanza aprendizaje se logre de manera fluida.

No sólo el público infantil se mostró interesado, también de otros rangos de edades lo externaron así, además de que alentaron a seguir haciendo este tipo de software.

Fue satisfactorio observar que el público infantil aprendió vocabulario en inglés y les sirvió de práctica para hacer sumas y restas sencillas, sin darse cuenta, ya que lo importante para ellos era el juego, sin embargo, se cumplió satisfactoriamente el objetivo de que la aplicación sirviera como auxiliar para aprendizaje aritmético e inglés, al brindar la opción bilingüe

Fue tal la aceptación entre la gran mayoría de usuarios que probaron el juego, que las sugerencias fueron en el sentido de añadirle más niveles de juego, otro tipo de vocabulario, otros tipos de niveles pero respetando la esencia de este software, videojuego, 3D y portátil.

Con esto, se pueden realizar trabajos futuros siguiendo este patrón para otras áreas de aprendizaje, como pueden ser Álgebra, Español, Alemán, Aritmética, etc., por lo que hay motivación suficiente para seguir trabajando de esta manera.

Bibliografía

- Blender. (01 de Enero de 2017). *“Primitives” Blender 2.79 Manual*. Recuperado el 16 de Septiembre de 2025, de Blender 2.79 Manual: <https://docs.blender.org/manual/es/2.79/modeling/meshes/primitives.html>
- Blender. (01 de Enero de 2017). *“Editing” Blender 2.79 Manual*. Recuperado el 18 de Septiembre de 2025, de Blender 2.79 Manual: <https://docs.blender.org/manual/es/2.79/modeling/meshes/editing/index.html>

Blender. (01 de Enero de 2017). “*Sculpting*” *Blender 2.79* Manual. Recuperado el 25 de Septiembre de 2025, de Blender 2.79 Manual: https://docs.blender.org/manual/es/2.79/sculpt_paint/sculpting/index.html

Blender. (01 de Enero de 2017). “*Armatures*” *Blender 2.79* Manual. Recuperado el 27 de Septiembre de 2025, de Blender 2.79 Manual: <https://docs.blender.org/manual/es/2.79/rigging/armatures/index.html>

Unity. (01 de Enero de 2019). “*Unity Essentials*”. Recuperado el 15 de Octubre de 2025 de Unity Learn: <https://learn.unity.com/pathway/unity-essentials>

Unity. (01 de Enero de 2019). “*Junior Programmer*”. Recuperado el 25 de Octubre de 2025, de Unity Learn: <https://learn.unity.com/pathway/junior-programmer>

Responsabilidades y funciones de los autores del documento.

Dr. Zárate Águila: idea original, investigación bibliográfica, captación y análisis de información, redacción del documento.

Dr. Águila Reyes: idea original, investigación bibliográfica, captación y análisis de información, redacción del documento.

Dr. González Tapia: idea Original, investigación bibliográfica, captación y análisis de información, redacción del documento.

Dr. Ascencio Sánchez: idea Original, investigación bibliográfica, captación y análisis de información, redacción del documento.