

## Desarrollo de videojuegos accesibles en Godot Engine: la problemática de los subtítulos

Federico Gabriel Pacheco, Ivana Harari, Claudia Queiruga

LINTI, Facultad de Informática, Universidad Nacional de La Plata (UNLP)  
La Plata, Buenos Aires, Argentina.

[federicocpacheco@gmail.com](mailto:federicocpacheco@gmail.com)

{iharari, claudiaq}@info.unlp.edu.ar

**Resumen.** En los últimos años se ha visto un incremento en la atención hacia la problemática de la accesibilidad dentro de la industria del videojuego. Los títulos lanzados por los grandes estudios productores de videojuegos evidencian un trabajo importante en torno a la accesibilidad de sus juegos. Si bien es cierto que en el sector independiente también se está produciendo un cambio de paradigma hacia un desarrollo más consciente de la problemática de la accesibilidad, las producciones que pertenecen a este sector de la industria son por lo general más vulnerables a barreras que impiden trabajar de manera adecuada la accesibilidad en sus juegos. En este artículo se pretende explicar cuáles son algunos de los problemas relacionados con la accesibilidad en Godot Engine específicamente los relacionados al subtítulo, y se propone una solución que permita utilizar subtítulos en los juegos creados con dicho motor.

**Palabras clave:** Videojuego, Accesibilidad, Subtítulos, Godot Engine, Plugin.

### 1 Introducción

La industria de desarrollo de videojuegos ha comenzado a experimentar un cambio importante durante los últimos años. Al igual que otros sectores dentro de la industria del entretenimiento, los videojuegos también comenzaron a ser más inclusivos, a tocar temáticas relacionadas a la conciencia social, a la identidad de género, y a la inclusión de personajes con diversidad cultural, religiosa y sexual en las historias y narrativas presentadas.

En particular la concientización acerca de la importancia de la accesibilidad de los juegos fue uno de los cambios más notables. Muchos de los juegos actuales incluyen, entre sus ajustes de configuración, opciones especialmente pensadas para brindar accesibilidad a los jugadores. Encontramos opciones de todo tipo: personalización de los controles y comandos de entrada, personalización de los colores y tamaños utilizados en la interfaz, soporte para ampliación de zonas de la pantalla, soporte para lectores de pantalla, soporte TTS<sup>1</sup>, y utilización de subtítulos. Todas estas

---

<sup>1</sup> Por las siglas en inglés de “Text to speech” (texto a voz).

características suponen un costo extra en el desarrollo de cualquier juego. Aunque estas son opciones que una gran mayoría de los jugadores nunca van a utilizar, no estar presentes significaría una barrera crítica en la interacción con el juego para una minoría. Una compañía de desarrollos de videojuegos grande puede poseer los recursos necesarios de tiempo, dinero y personal capacitado para poder llevar adelante la implementación de estrategias de accesibilidad en sus juegos. Por el contrario, para un estudio independiente ésto puede requerir la utilización de recursos que no se poseen, o no se están dispuestos a invertir (Knight Z., 2011).

Una de las herramientas para la creación de videojuegos más utilizada por los desarrolladores de videojuegos independientes es Godot Engine, un motor de videojuegos de software libre, desarrollado en Argentina. De acuerdo a la última encuesta realizada a los desarrolladores que utilizan Godot (Godot Community Poll, 2021) podemos identificar que esta comunidad por lo general utiliza Godot para realizar juegos de manera independiente, es decir que no trabajan para una compañía. Esta encuesta nos dice también que los usuarios de Godot por lo general trabajan solos en la realización de sus juegos. Este motor cuenta en la actualidad con un conjunto muy completo de herramientas para crear videojuegos, aún así no brinda soluciones específicamente diseñadas para trabajar la accesibilidad. Todas las herramientas que ofrece Godot soportan algún tipo de personalización, desde los componentes de la GUI (Graphical User Interface), los controles, el motor de física, los efectos de sonido, etc. Esto deja abierto el camino para que el desarrollador exponga esta personalización de los componentes al jugador, generando accesibilidad. En Godot es el desarrollador quién debe encargarse íntegramente de brindar soporte de accesibilidad en los juegos. Por ejemplo, aunque se permite mostrar en pantalla imágenes y texto, en diferentes colores y tamaños, no se ofrece una solución concreta para crear subtítulos, quedando la implementación de estos a cargo del desarrollador. Más allá del grado de dificultad que conlleva implementar estrategias de accesibilidad para los juegos hechos en Godot, es importante para la comunidad de desarrolladores independientes contar con la mayor cantidad posible de herramientas y opciones enfocadas en crear accesibilidad de una forma sencilla y rápida dentro del motor. Esto no solo es una solución para el desarrollador independiente, que no tiene los recursos para implementar accesibilidad en sus juegos, sino que también es un beneficio para todos los jugadores que necesitan este tipo de soluciones de accesibilidad.

## **2 Diversidad de jugadores**

De acuerdo a un informe de DFC Intelligence, un proveedor de análisis de la industria del videojuegos, alrededor de 40% de las personas a nivel mundial juegan videojuegos (Global Video Game Consumer Segmentation, 2020). Esto equivale a más de 3 mil millones de personas, siendo 383 millones de América Latina. En el caso particular de la Argentina, según un estudio publicado por Newzoo sobre jugadores argentinos, existen 19 millones de argentinos jugando videojuegos móviles, esto equivale a aproximadamente 42% de la población total (The Argentinian Gamer,

2017). Del mismo estudio también podemos estimar que el 57% de los jugadores son hombres. Y de acuerdo al rango etario podemos decir que la mayor cantidad de jugadores se encuentra en el rango de edad de 21 a 35 años (40%), seguido por los jugadores de entre 10 y 20 años (33%), luego por los jugadores de entre 35 y 50 años (22%), y luego los jugadores de más de 51 años (5%).

El estudio relacionado a discapacidades realizado por el INDEC indica que en Argentina el 10,2% de la población tiene algún tipo de discapacidad, lo cual equivale a casi 5 millones de personas (Estudio Nacional sobre el Perfil de las Personas con Discapacidad, 2018). De este estudio también podemos realizar un análisis por rango etario. En la población de 6 a 14 años de edad predominan las dificultades mental-cognitivas (48.3%). En el grupo de 15 a 65 años predominan las personas con deterioro motriz y visual. Y en el rango de edad mayor a los 65 años, predominan las personas con discapacidades motrices. Separando los datos por tipo de discapacidad vemos que la dificultad motriz es la principal causa de discapacidad con un 4,9%, seguido de la discapacidad visual con un 2,5%, la discapacidad auditiva con un 2,1%, dificultades para entender, aprender, recordar o concentrarse con 1,4% y por último las dificultades en el habla y la comunicación con 0,8%.

Analizando los datos arrojados por estos estudios podemos decir, más allá de mirar los datos en crudo, que muchos argentinos juegan videojuegos, sin importar su edad o género. También podemos ver que las dificultades provocadas por las discapacidades están presentes en absolutamente todos los rangos de edad. La suposición general de que las personas con discapacidad representan un pequeño grupo dentro del mercado de los videojuegos puede indicar que la falta de accesibilidad en los videojuegos son por lo general un resultado de la omisión y de la falta de atención más que de un diseño deliberado (Ellis y Kao, 2019). Generalmente se considera a los videojuegos como una mera forma de entretenimiento, sin embargo para un jugador con discapacidad el videojuego puede cobrar otra relevancia, permitiéndole conectar con otras personas, relajarse y divertirse con amigos interactuando en un ambiente seguro. Adicionalmente, los videojuegos pueden ser utilizados como herramientas con fines educacionales y terapéuticos. Francesca Gauci, por ejemplo, presenta en su tesis (*Game Accessibility for Children with Cognitive Disabilities*, 2021) una serie de soluciones de accesibilidad para mejorar la interacción entre los videojuegos y los jugadores con dificultades cognitivas como el trastorno de aprendizaje, trastorno de la coordinación motora y el trastorno del déficit de la atención. Francesca trabaja su tesis desde un punto de vista terapéutico, y analizando a los juegos “serios”, es decir los juegos que son creados con una finalidad diferente al entretenimiento. Pero más allá de las posibilidades en tratamientos y rehabilitación es importante eliminar las barreras entre los jugadores con diversidad funcional y cualquier tipo de videojuego. Las personas con discapacidad también juegan, y como cualquier otra persona, tienen derecho de acceder a este tipo de entretenimiento. Otros tipos de entretenimientos están adelantados varios años en cuanto a la inclusión legal de personas con discapacidad: las películas, los libros, las salas de cine, los deportes, etc. En los videojuegos todavía no existe esta mirada, por lo que los desarrolladores tienen el

deber, aunque no haya regulaciones legales fuertes todavía, de realizar sus productos para una diversidad de jugadores cada vez más grande.

### **3 Videojuegos accesibles**

Cuando los juegos imponen limitaciones sobre los jugadores con discapacidad, al no ofrecerles una manera de interactuar adecuadamente con el juego, obstaculizan la capacidad del jugador para sumergirse en la experiencia completa del “juego” como actividad libre y absorbente (Huizinga, 1955). Es más, el trabajo extra que deben realizar los jugadores con discapacidad al tratar de romper esos obstáculos que encuentran al interactuar con el juego representa una barrera que los jugadores deben sortear y que por lo general les demanda un esfuerzo y costo adicional.

Por lo general los jugadores encuentran su propia solución alternativa para sortear estas barreras en los juegos. Por ejemplo, los jugadores con dificultades visuales se apoyan en los elementos de sonido del juego, en la ayuda externa de un jugador mediante un chat de voz, o ajustan los colores y tamaños de los elementos visuales desde las opciones de configuración del juego para hacerlas más visibles. Dependiendo de la naturaleza del juego y de las necesidades del jugador, algunas de estas soluciones alternativas pueden llegar a ser tan complejas como el remapeo del control del juego para aceptar ser controlado con los pies, o la utilización de extensiones de terceros para poder controlar el juego usando la mirada del jugador.

Por lo general consideramos a los videojuegos como una mera forma de entretenimiento, sin embargo para un jugador con discapacidad el videojuego puede cobrar otra relevancia, permitiéndole conectar con otras personas, o relajarse y divertirse con amigos interactuando en un ambiente seguro. Adicionalmente los videojuegos pueden ser utilizados como herramientas con fines educacionales y terapéuticos. Pueden ser una herramienta poderosa para mejorar la conectividad cerebral, mejorar el tiempo de respuesta, la atención, la navegación espacial e incluso la memoria de corto plazo en adultos mayores (Anguera y Gazzaley, 2015). Solo porque una persona tiene una dificultad o necesidad especial no significa que debe ser excluida y quitarle la posibilidad de jugar y aprender.

El objetivo de los desarrolladores de videojuegos entonces es crear videojuegos accesibles, permitiendo a los jugadores con discapacidad reducir la mayor cantidad posible de esfuerzo adicional requerido para interactuar con el juego. Esto significa analizar qué tipo de dificultades pueden surgir para personas con discapacidad de acuerdo al tipo de juego que estamos creando, y diseñar estrategias que reduzcan estas dificultades para el jugador. Estas estrategias puede que nunca lleguen a alcanzar a la gran mayoría de los jugadores, y podemos caer en el error de pensar que por eso no son importantes, pero para la minoría que realmente las necesita tiene un impacto realmente tangible. El hecho de que un juego soporte o no ciertas opciones de accesibilidad puede hacer que un jugador opte por no jugarlo, o puede que el juego sea lo suficientemente inaccesible para excluir al jugador por completo.

## 4 Godot Engine

Godot Engine (<https://godotengine.org/>) es un motor de videojuegos de uso general, de código abierto y publicado bajo la licencia MIT. Permite crear videojuegos, aplicaciones de escritorio y para dispositivos móviles de una forma sencilla, a través de un IDE propio desde el cual se accede a todas las herramientas que ofrece. Las aplicaciones creadas con Godot se pueden exportar a diversas plataformas como Windows, Linux, Mac OS, dispositivos Android y iOS, y consolas de juegos como Xbox y PlayStation. Uno de los pilares de Godot es el diseño simple, extensible y accesible. Esto lo podemos ver en muchos de los aspectos que rodean a este proyecto: la colaboración, la arquitectura del motor, el diseño de la UI, la manera de utilizar nodos y escenas como componentes reutilizables, y en GDScript: el lenguaje de scripting creado cuidadosamente para ser de fácil aprendizaje y utilización. Godot Engine es utilizado en mayor parte por estudios y desarrolladores independientes (Godot Community Poll, 2021) para crear videojuegos como: “Ex-Zodiac”, un juego de disparos sobre rieles de estilo low poly lanzado en varias plataformas en el año 2022, y “Haiki” un juego de plataformas en 2D lanzado en 2021.

### 4.1 Accesibilidad en Godot Engine

Quizás una de las características más inteligentes que tiene Godot es que su IDE está creado utilizando Godot mismo. Al contrario que en otros motores de videojuegos, donde la GUI del editor está construida sobre una librería totalmente diferente, en Godot se utilizan los componentes de la UI tanto para armar juegos como para el editor mismo. Podemos destacar que las mejoras y los avances que se realicen sobre los componentes y widgets del editor también quedarán disponibles para los desarrolladores que utilicen estos mismos componentes en sus juegos. Por ejemplo, si se soluciona un bug en un componente de input de texto del editor, este bug estará solucionado también en el input de texto dentro del juego. Esto trae ciertas ventajas ya que permite utilizar el mismo código y componentes tanto para crear un juego como para extender el editor. Además significa que los componentes de la interfaz del editor están disponibles para todas las plataformas en las que Godot está disponible, facilitando la portabilidad del editor a otras plataformas. El hecho de implementar una librería de GUI trae consigo sus propias desventajas, siendo una de las más grandes la responsabilidad de implementar comportamientos comunes que podemos esperar de los componentes y widgets, como por ejemplo la navegación y el uso de atajos de teclado, soporte para pantallas táctiles, y por supuesto las opciones de accesibilidad. Si bien en Godot la GUI es robusta y está bastante desarrollada, no es el aspecto que genera más interés dentro de un motor de videojuegos en crecimiento, por lo que el soporte de accesibilidad queda relegado ante otros desarrollos más “importantes”. Con el tiempo algunos colaboradores comenzaron a recopilar problemas y se crearon algunas propuestas para mejorar Godot en este aspecto. Algunos desarrolladores

incluso crearon plugins no oficiales que mejoran el soporte de accesibilidad de los componentes.

Un ejemplo de esto es el plugin “Godot Accessibility” creado por Nolan Darilek, el cual incorpora mejoras de accesibilidad al editor de Godot. Este plugin agrega capacidad TTS a Godot, y la implementación de un lector de pantalla integrado. La lectura de pantalla está implementada mediante un nodo llamado “ScreenReader”, el cual contiene la lógica necesaria para transformar las interfaces visuales de Godot en algo accesible y navegable de una forma similar a la que se hace con un lector de pantalla tradicional. El plugin incluye mejoras para que la navegación de la UI sea accesible mediante el teclado y mediante gestos comunes utilizados en lectores tradicionales. Además intenta solucionar varios problemas relacionados al manejo de eventos presentes en Godot. Por ejemplo, intenta hacer que los elementos de UI tomen foco automáticamente cuando una escena es inicializada. El plugin también intercepta automáticamente todos los eventos de entrada de las pantallas táctiles para emular el funcionamiento de la navegación en lectores de pantalla al estilo Android y iOS, donde se navega utilizando los gestos “swipe” y “double tap”. El plugin también expone al desarrollador una interfaz para interactuar con el sistema operativo y utilizar sus capacidades de texto a voz. Esto es utilizado por el plugin para poder leer en voz alta el contenido de varios de los componentes de UI más comunes.

#### **4.2 Punto de vista del desarrollador**

Ya vimos que es importante pensar en la accesibilidad al momento de diseñar nuestro software, ya sea un videojuego u otro producto de software. En el caso de Godot, además de pensar en crear videojuegos accesibles con él, podemos revisar su accesibilidad y examinar qué tan accesible es para con los desarrolladores que lo utilizan. Una de las falencias más grandes de Godot es que la interfaz del editor no es compatible con lectores de pantalla, siento una barrera para muchos desarrolladores que utilizan este tipo de tecnología. El plugin de Nolan soluciona este problema, inyectando el nodo “ScreenReader” dentro del editor. Esto habilita, en teoría, la navegación mediante lector de pantalla en todo el editor, utilizando los controles comunes “Tab” y “Shift + Tab” en conjunto con las teclas de dirección del teclado para navegar entre las diferentes secciones del editor, y además pronunciando en voz alta el nombre y contenido del componente cada vez que toma el foco. Es un avance importante en soporte de la accesibilidad de la GUI y vuelve al IDE mucho más amigable a los usuarios con capacidades visuales disminuídas.

Por desgracia, siguen existiendo algunos problemas en la interfaz del editor que son difíciles de solucionar mediante el uso de plugins. Estos problemas están ligados al funcionamiento de los componentes mismos y del diseño del editor, por lo que deben ser solucionados por la implementación del motor en sí y no mediante plugins.

Uno de los problemas más importantes es que el editor mismo es demasiado complejo como para desarrollar un juego completo sin la utilización del mouse.

Existen muchas secciones dentro del editor donde la navegación mediante el teclado es complicada, o simplemente imposible. La vista del sistema de archivos y el editor de scripts son dos de las áreas particularmente complicadas de controlar. Algunos de los controles son muy complejos para manipular (incluso utilizando el mouse) como por ejemplo el editor de gradientes, el editor de árbol de animaciones y el editor de polígonos. Para algunos de estos controles se permite la edición mediante el ingreso de valores por teclado, como por ejemplo para la edición de gradientes. La alternativa para estos controles altamente visuales, es una edición manual de los archivos de configuración en un software externo, algo que se podría evitar ofreciendo un diseño más inclusivo.

Uno de los aspectos más difíciles de la navegación vía teclado es la pérdida del foco de los componentes. Esto es un problema que se fue mejorando en las nuevas versiones, pero que no está completamente resuelto. La navegación de la interfaz vía teclado depende de que los componentes de la UI no pierdan el foco, algo que sucede con frecuencia dentro del editor. Un ejemplo de este problema se da cuando se lanza la preview del juego: cuando se abre la ventana del juego, el editor pierde el foco, y no lo obtiene incluso cuando la ventana del juego se cierra. Otro ejemplo podemos verlo cuando navegamos por el inspector de propiedades: una vez que navegamos hasta una propiedad de ingreso de texto en modo multilínea (por ejemplo editando un componente Label), las combinaciones de teclado “Tab” y “Shift + Tab” que se usan para navegar entre componentes dejan de tener efecto, y comienzan a usarse para ingresar el carácter de control “Tab” dentro del texto. Para poder ganar el control de la navegación vía teclado es necesario utilizar el mouse para clickear fuera del componente.

### **4.3 Punto de vista del jugador**

No todo es malo en cuanto a la accesibilidad en Godot. Como aspecto positivo podemos mencionar que a medida que fue creciendo la adopción de este motor dentro de la comunidad de desarrolladores, se comenzó a prestar más atención a algunos puntos básicos sobre la accesibilidad. Las nuevas versiones del motor permiten personalizar la configuración del tema que utilizan todos los componentes y widgets, lo cual permite a los jugadores con dificultades visuales cambiar el esquema de colores por uno que se adecuen a sus necesidades.

En cuanto al texto, se permite cambiar la tipografía, color y tamaño de las fuentes. Los jugadores con dificultad visual más grave pueden aplicar un factor de escala para la UI, generando un efecto de ampliación sobre la interfaz y facilitando la identificación de los elementos en pantalla. Adicionalmente se permite cambiar el tamaño de espacios, barras de división y otros elementos que facilitan la separación e identificación de las secciones y componentes de las interfaces dentro del juego.

Una de las mejoras más útiles para los jugadores con discapacidades motrices, es la posibilidad de crear atajos para interactuar con algunos componentes de la UI. Esto

permite, por ejemplo a un jugador con problemas motrices, crear atajos para realizar tareas repetitivas en la UI, o acceder rápidamente a un menú, eliminando barreras en la utilización de las herramientas. Lo remarcable de esta mejora es que se pueden crear atajos a partir de cualquier tipo de evento: teclado, gestos de pantalla, joystick, dispositivos MIDI, etc. Esto abre la posibilidad de controlar e interactuar con la UI utilizando dispositivos externos como por ejemplo teclados adaptativos o interruptores “sip and puff”.

Debemos recordar que los componentes de GUI con los que interactúan los jugadores en los juegos son los mismos que utilizan los desarrolladores dentro del IDE, por lo tanto los conceptos, ventajas y desventajas vistos desde el punto de vista del jugador, también aplican para las interfaces utilizadas dentro del editor y viceversa. Por ejemplo, en el caso del plugin “Godot Accessibility”, las capacidades de lector de pantalla que se introducen al editor, también están disponibles para los jugadores cuando interactúan con un juego creado con ese plugin.

Todas estas opciones de personalización que brinda Godot no brindan accesibilidad por sí mismas, sino que es responsabilidad del desarrollador hacer un uso correcto de las mismas. Por lo general se espera que el desarrollador incluya un menú de opciones dentro del juego, donde se expongan estas configuraciones al jugador: personalización de los colores y tamaño de las interfaces, cambio de controles, creación de atajos, etc.

## 5 Problemáticas de subtítulos identificadas en Godot Engine

El motor Godot Engine no da soporte oficial a la creación y manejo de subtítulos dentro del motor. De no existir una herramienta oficial para la creación de subtítulos, un grupo de desarrollo pequeño, con tiempos y recursos limitados, podría encontrar una barrera a la hora de considerar integrar subtítulos en sus juegos. Esta es una práctica común en muchos videojuegos del sector independiente, con efectos que tienden a la exclusión de parte de la comunidad de jugadores que necesitan los subtítulos para poder jugar. A continuación se identifican algunas de las problemáticas más comunes que pueden derivar de la falta de soporte de subtítulos dentro de Godot Engine.

### 5.1 Falta de subtítulos descriptivos

**Descripción:** El juego no ofrece subtítulos descriptivos, o “closed-caption” para describir eventos de índole auditiva importantes para el jugador.

**¿A qué jugadores afecta?** Los jugadores con deterioro auditivo parcial o total pueden necesitar una alternativa para mejorar la percepción de la información visual o auditiva presentada por el juego.

**Problema:** Este problema está dado no por la falta de subtítulos sobre los diálogos de los personajes, sino que se enfoca en la falta de subtítulos sobre cualquier tipo de evento exclusivamente auditivo que, aunque no es esencial para el jugador, sí es



importante para procesar y entender el contexto dentro del juego. En este grupo podemos incluir cualquier tipo de sonido que se utiliza para enfatizar o comunicar información importante para el jugador: disparos, sonidos de alarmas, alertas, sonidos del ambiente, sonidos out (fuera de campo de visión), golpes, etc. Estos tipos de sonido terminan de completar el contexto del juego, y la falta de subtítulos para los mismos puede constituirse en una barrera para los jugadores con deterioro auditivo. Por ejemplo, imaginemos un juego donde tenemos una palanca que abre una puerta que el jugador debe cruzar para poder avanzar. Al utilizar la palanca, vemos que la puerta se abre, y escuchamos el sonido “tic-tac” de un reloj. Un jugador que escucha ese sonido entiende entonces que la puerta se va a mantener abierta solo por un período de tiempo corto, y que luego se cerrará, por lo que deberá acelerar el paso para poder cruzarla. Un jugador con deterioro auditivo no escuchará este sonido y por lo tanto no comprenderá el cambio de contexto del juego, por lo que no acelerará el paso, y encontrará la puerta cerrada. Esto puede llegar a ser confuso o hasta frustrante para los jugadores.

**Solución:** Los subtítulos “closed caption” o subtítulos cerrados se utilizan en este caso para describir esta información contextual de forma visual mediante un texto o descripción que acompaña al sonido. En el ejemplo de la palanca y la puerta, si mostramos en pantalla un subtítulo indicando que “se escucha el tic-toc de un reloj” cuando se activa la palanca, todos los jugadores (hayan escuchado el sonido o no) entenderán que deben apurarse.

**Beneficios:** Este tipo de subtítulos es muy útil para brindar información contextual al jugador, creando una inmersión mucho más profunda, y evitando barreras en el juego. Adicionalmente estamos creando un diseño más robusto, ya que permite tanto a jugadores que posean algún deterioro auditivo como a jugadores que simplemente les gusta jugar con el sonido apagado, una experiencia de juego similar. También podemos aplicar este principio a otros elementos que aportan contexto dentro del juego, como por ejemplo elementos interactivos, textos y escritos dentro del mundo del juego que pueden ser de difícil lectura (cartas, notas, graffitis en paredes), lugares o personajes. En la figura 1 vemos una captura de Portal 2, juego que utiliza subtítulos descriptivos para los sonidos de las armas, enemigos y sonidos del ambiente, como por ejemplo los sonidos que hacen las puertas al abrirse y cerrarse.



**Fig. 1.** Captura de pantalla del juego Portal 2 (2011), donde se utilizan subtítulos descriptivos para los sonidos de las puertas y otros sonidos del ambiente.

## 5.2 Problema para identificar al hablante

**Descripción:** El jugador encuentra problemas para identificar el personaje que está hablando

**¿A qué jugadores afecta?** Los jugadores con dificultades visuales o cognitivas pueden necesitar un refuerzo adicional para poder determinar el origen de las líneas de diálogo.

**Problema:** En los casos donde el jugador tiene problemas para escuchar el doblaje de los diálogos, no entiende el idioma o los acentos de los personajes, se puede crear una barrera en donde el jugador no identifica el origen de las líneas de diálogo. Adicionalmente en algunos tipos de juegos, como los RPGs, aparecen muchos personajes en pantalla a la vez, esto también puede ser un problema para identificar cuáles de los personajes es el que está hablando.

**Solución:** De acuerdo a especificaciones técnicas sobre la implementación de subtítulos creada por Microsoft para los juegos en Xbox (XBox Accessibility Guidelines), los subtítulos deben incluir el nombre del personaje que está hablando al inicio de la línea de diálogo. Esto debe mostrarse solo la primera vez que habla el personaje, pudiendo removerse en las líneas subsecuentes. También se debe incluir el nombre en casos donde el diálogo se retoma después de ser interrumpido por el diálogo de otro personaje. Adicionalmente se utiliza el color para representar el vínculo con el personaje, donde cada personaje tiene su color característico. Otras opciones de vínculo con el personaje que se pueden utilizar son por ejemplo un

indicador en la dirección donde está el personaje que habla, o la utilización de burbujas de diálogo.

**Beneficios:** Para los jugadores con deterioro auditivo completo puede ser crucial tener la posibilidad de identificar el personaje por el nombre o el color, sobre todo en juegos fuertemente basados en diálogos como en las aventuras gráficas. Adicionalmente ofrecer a los jugadores un control más fino sobre el tipo de colores elegidos para los personajes, el tamaño y posición de los subtítulos puede ayudar a jugadores con algún tipo de deterioro visual, sobre todo los que poseen baja visión y trastornos de la percepción del color a utilizar más cómodamente el juego. Algunos juegos muestran los subtítulos cerca de los personajes cuando hablan, una práctica que puede ayudar en la identificación del personaje, pero puede ocasionar otro tipo de problemas en jugadores con baja visión, sobre todo cuando esperan que el diálogo aparezca como lo hace normalmente en el área central baja de la pantalla.

## 5.2 Cinemáticas sin subtítulos

**Descripción:** El juego incluye subtítulos pre-renderizados o no incluye ningún tipo de subtítulos para las cinemáticas.

**¿A qué jugadores afecta?** Este tipo de problemática puede afectar tanto a los jugadores con deterioro auditivo, como aquellos con deterioro visual.

**Problema:** Las cinemáticas son esencialmente elementos multimedia que utilizan el audio y el sonido para contar una historia o narrativa. Se implementa generalmente como videoclip o archivo de película que se proyecta en la pantalla, momento en el cual el jugador no tiene control sobre el juego y solo se espera que se sumerja en la historia presentada. En el caso de Godot, donde no existe un soporte oficial para los subtítulos, podemos encontrar gran variedad de juegos que muestran cinemáticas donde no se incluyen subtítulos, excluyendo a muchos jugadores que no pueden escuchar el diálogo doblado en pantalla. Esto se suele solucionar de una forma “barata” pre-renderizando los subtítulos sobre el video. Esto es una práctica que tiene varios problemas en sí misma: estos subtítulos no soportan traducción a otros idiomas, ni el ajuste de tamaño y color, por tratarse de texto renderizado en el video.

**Solución:** Utilizar subtítulos in-game para todo nuestro juego, incluyendo las cinemáticas. Usar subtítulos in-game en lugar de pre-renderizados nos permite además controlar el aspecto visual del mismo: tamaño, tipografía, posición, etc.

**Beneficios:** Los jugadores que poseen un deterioro auditivo completo pueden disfrutar, al igual que el resto de los jugadores, de los momentos donde se desarrolla la historia del juego. Además permite a los jugadores con baja visión poder ajustar el aspecto de los subtítulos durante las cinemáticas. Incluso los jugadores que no entienden el lenguaje en el que se desarrolla la cinemática pueden encontrar en los subtítulos la accesibilidad que necesitan para acceder a la historia del juego.

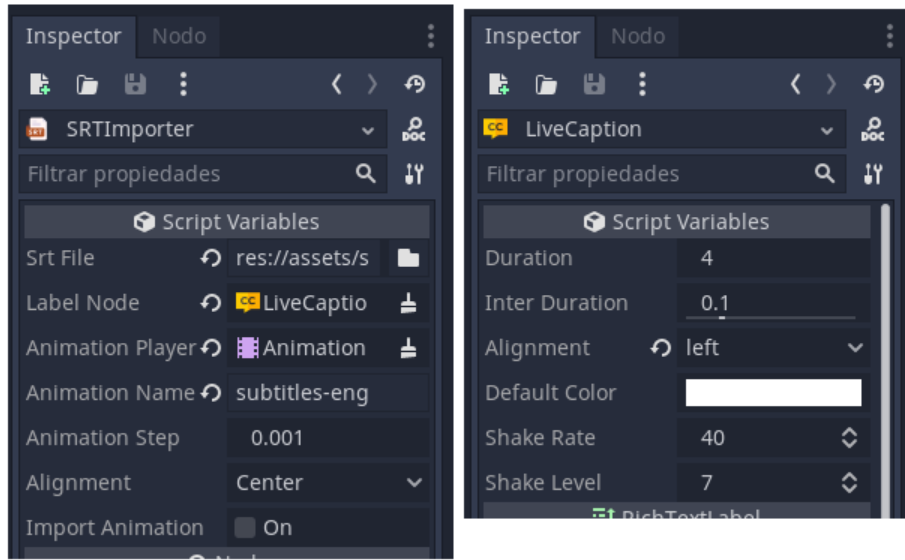
## 6 Plugin para crear subtítulos en Godot

Según algunos colaboradores, las herramientas que provee el motor para mostrar texto en pantalla deberían ser suficientes para poder implementar subtítulos manualmente dentro del juego (Godot Engine Accessibility Proposals). Aunque esto es verdad, no deja de ser cierto que la falta de soporte oficial para la creación de subtítulos influye tremendamente en la ausencia de subtítulos en los juegos independientes creados en Godot. Por este motivo resulta relevante para este trabajo tomar la responsabilidad de crear un plugin que permita de forma sencilla incluir subtítulos en las aplicaciones y juegos creados con Godot.

Godot Engine está diseñado de manera que las funcionalidades y herramientas sean fácilmente extensibles. Al ser un proyecto de software libre, siempre se puede cambiar el código fuente original y agregar el soporte que el juego necesita. En particular Godot provee tres maneras diferentes de agregar nueva funcionalidad al motor: mediante un nuevo módulo en el core del motor, utilizando “GDNative”, o mediante la creación de plugins. La primera opción se basa en agregar la lógica necesaria mediante un nuevo módulo y compilar el código fuente del motor entero. GDNative es una tecnología específica de Godot Engine que permite crear módulos en forma de librerías que se pueden anexas en tiempo de ejecución al ejecutable del juego. Por último se permite extender el motor mediante plugins, una forma que facilita mucho su distribución ya que para que funcionen solo se deben incluir dentro del proyecto original a modo de add-on, y su código no debe ser compilado lo que permite ser compatible con cualquier plataforma en donde se ejecute.

El plugin desarrollado se denomina “Github Subtitles Plugin”, está desarrollado íntegramente con GDScript, lo que facilita su extensión a futuro y compatibilidad con nuevas versiones de Godot. No necesita soporte nativo de la plataforma donde se ejecuta el juego, y se puede instalar de forma sencilla en cualquier proyecto. Introduce dos nuevos nodos que brindan soporte para subtítulos en Godot: “SRTImporter” y “LiveCaption”. El primero se centra en brindar soporte para leer archivos de subtítulos con formato srt y convertirlos a un formato que se pueda utilizar dentro del motor. El nodo “SRTImporter” es un nodo utilitario que está pensado para ser usado solo durante el desarrollo del juego, y no durante la ejecución. Permite al desarrollador seleccionar un archivo srt, parsear el contenido de acuerdo a las especificaciones del formato SRT, y guardarlo como una animación, un recurso que se almacena en disco y que se puede reproducir en tiempo de ejecución con un AnimationPlayer. La animación solo contiene información básica: el texto que se quiere mostrar, alineación y formato básico (negrita, subrayado, cursiva, etc). Esto deja otros atributos importantes bajo el control del desarrollador: la tipografía, colores, tamaños, y espaciado de la fuente se pueden configurar para encajar con el diseño del juego o para hacerlo más accesible. Como vemos en la figura 2, el nodo permite seleccionar desde el disco un archivo de subtítulos en formato srt, un nodo que permita mostrar el texto, y un nodo en donde almacenar la nueva animación. Adicionalmente se permite configurar la resolución de la animación, y aplicar una alineación a los subtítulos (centrado, izquierda, derecha). Este nodo se puede utilizar

para solucionar por ejemplo la problemática de las cinemáticas sin subtítulos, solo basta con importar el archivo de subtítulos y reproducirlo en conjunto con la cinemática.

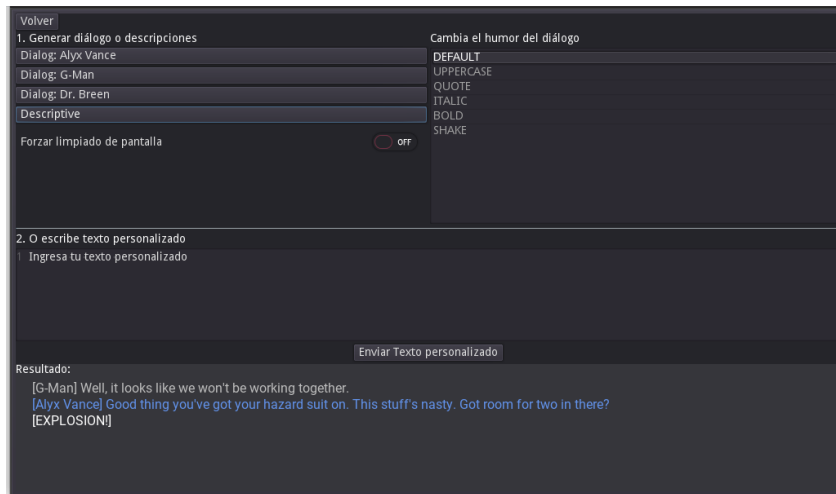


**Fig. 2.** Vista de los atributos configurables del nodo SRTImporter (izquierda) y el nodo LiveCaption (derecha).

El segundo nodo, llamado “LiveCaption”, permite generar subtítulos dinámicamente, y está pensado para usarse en tiempo de ejecución. La idea de este nodo es mostrar en pantalla subtítulos sólo cuando el desarrollador lo indica, no depende de ninguna información previa. Se puede utilizar para mostrar en pantalla el diálogo de los personajes, o para generar subtítulos descriptivos ante eventos dinámicos dentro del juego. Este plugin soporta la utilización de colores y nombres de personajes con cada línea de diálogo, los cuales muestra en pantalla cuando está indicado por el desarrollador. Como se indica en la guía “XBox Accessibility Guidelines”, el nombre del personaje solo se muestra la primera vez que habla, y se oculta en las siguientes líneas de diálogo. Las líneas de diálogo permanecen en pantalla por un tiempo configurable, antes de desaparecer. En la figura 2 podemos ver la configuración para el nodo “LiveCaption”, permitiendo cambiar la duración de cada línea de diálogo en pantalla, el espacio de tiempo entre subtítulos, forzar una alineación (centrado, izquierda, derecha), el color por defecto de los subtítulos, y la configuración del efecto “shake” (uno de los efectos que se pueden aplicar a los textos en Godot).

En el sitio de GitHub (<https://github.com/FEDEOD/godot-subtitles>) se encuentra disponible un proyecto que ejemplifica cómo utilizar el plugin “Godot Subtitles

Plugin” para resolver las problemáticas planteadas en relación a los subtítulos en Godot. Este proyecto incluye una demostración de cómo subtítular un video usando el plugin para importar un archivo de subtítulos en formato srt. El proyecto también incluye un generador de diálogos (figura 3), el cual utiliza el nodo “LiveCaption” para ejemplificar la utilización de colores y nombres de los personajes en los diálogos, además de la generación de subtítulos descriptivos.



**Fig 3.** Generador de diálogo y subtítulos descriptivos que se incluyen en el proyecto de ejemplo, ejemplificando el uso del plugin “Godot Subtitles Plugin”.

## 7 Conclusiones

Los subtítulos son una de las estrategias básicas para la creación de accesibilidad en cualquier tipo de recurso multimedia, y como tal es importante que los desarrolladores de videojuegos que utilizan Godot puedan incluirlos en sus juegos. En este artículo se visibiliza la problemática de los subtítulos presente en Godot Engine, y se trata de demostrar qué efectos puede tener sobre la comunidad de jugadores. Se evidencia la importancia de la correcta implementación de subtítulos en los videojuegos mediante la identificación de problemáticas, y se muestran posibles soluciones a estos problemas de accesibilidad.

Godot Engine presenta varios problemas que dificultan la creación de juegos y aplicaciones accesibles, impactando tanto en jugadores como en los desarrolladores que utilizan el motor. Siendo que algunas de las herramientas disponibles dentro del motor no facilitan la implementación de estrategias de accesibilidad, queda bajo la responsabilidad del equipo de desarrollo implementar dichas estrategias. La falta de recursos económicos y otras dificultades en el desarrollo que acompañan a los equipos que por lo general usan este motor, atentan contra la implementación de estas estrategias, con lo cual es de suma importancia para la comunidad de Godot conocer

las problemáticas que surgen de la falta de accesibilidad, y proveer mejoras sobre las herramientas oficiales que faciliten la implementación de la misma.

## Referencias

- Anguera J., Gazzaley A. (2015): Video games, cognitive exercises, and the enhancement of cognitive abilities.
- Darilek N., Godot Accessibility Github Repository webpage, <https://github.com/lightsoutgames/godot-accessibility>, last accessed 2022/12/06
- DFC Intelligence. (2020) Global Video Game Consumer Segmentation.
- Ellis K., Kao, K. (2019): Who Gets to Play? Disability, Open Literacy, Gaming. Ex-Zodiac Steam store webpage, <https://store.steampowered.com/app/1249480/ExZodiac/> last accessed 2022/19/08
- Game Accessibility Guidelines website, <https://gameaccessibilityguidelines.com/>, last accessed 2022/12/06
- Gauci, F. 2021. Game Accessibility for Children with Cognitive Disabilities
- Godot Engine website, <https://godotengine.org/>, last accessed 2022/12/06
- Godot Community Poll 2021, [https://docs.google.com/forms/d/e/1FAIpQLSfOj9m71X98\\_z7-UC4iiM6Uxd2HhITRzmF-M2Yx-d7Yq5KNUg/viewanalytics](https://docs.google.com/forms/d/e/1FAIpQLSfOj9m71X98_z7-UC4iiM6Uxd2HhITRzmF-M2Yx-d7Yq5KNUg/viewanalytics), last accessed 2022/12/06
- Godot Engine Accessibility Proposals Github Issues webpage, <https://github.com/godotengine/godot-proposals/issues/983>, last accessed 2022/12/06
- Haiki Steam store webpage, <https://store.steampowered.com/app/1395270/Haiki/> last accessed 2022/19/08
- Huizinga, J. (1955). Homo ludens: A study of the play-element in culture.
- INDEC. Agencia Nacional de Discapacidad. (2018): Estudio Nacional sobre el Perfil de las Personas con Discapacidad
- Knight, Z. (2011): Random Battles: A Gamer's Guide to What the Crap is Happening in the Games Industry.
- Mangiron C., Orero P., O'Hagan M. (2014): Fun For All. Translation and Accessibility Practices in Video Games.
- NewZoo (2017). The Argentinian Gamer.
- Pacheco F., Godot Subtitles Plugin Github Repository webpage, <https://github.com/FEDE0D/godot-subtitles>, last accessed 2022/12/06
- Shin M. (2021): A Growth of Accessibility in Video Games. Disabilities, Opportunities, Internetworking, and Technology. University of Washington.
- XBox Accessibility Guidelines, <https://docs.microsoft.com/en-us/gaming/accessibility/guidelines>, last accessed 2022/12/06