

Propuesta inicial del videojuego

Juan Felipe Orozco Londoño - Maria Valentina Quiroga Alzate

Informática II – 2025_1

Ingeniería Electrónica

1. Capítulos escogidos:

Episodios #70 al #76 de Dragon Ball
Saga: El Palacio de Uranai Baba

Justificación: Esta saga fue elegida por su variedad de combates creativos, personajes y escenarios diferentes que permiten diseñar niveles entretenidos y adaptables a un videojuego sencillo con físicas implementables en Qt.

2. Contextualización de los capítulos

Los episodios elegidos componen la mini-saga del Palacio de Uranai Baba. Esta saga se sitúa justo después de la derrota del Ejército de la Patrulla Roja, cuando Gokú busca la última esfera del dragón, pero descubre que el radar no puede detectar su ubicación, por lo que acude al palacio de Uranai Baba, una adivina misteriosa y excéntrica que acepta ayudarlo solo si él y sus amigos logran derrotar a cinco guerreros sobrenaturales que ella misma invoca.

La saga del Palacio de Uranai Baba, presenta una secuencia de desafíos únicos que combinan acción, humor y elementos sobrenaturales, ideal para ser adaptada al videojuego.

Todo comienza en los capítulos 70 y 71, cuando Gokú y sus amigos llegan al palacio de Uranai Baba en busca de la última esfera del dragón. Sin embargo, la adivina impone la condición de que antes de revelar la ubicación de la esfera, deberán enfrentarse a cinco guerreros misteriosos invocados por ella misma. A continuación, en el capítulo 72, Yamcha se enfrenta al primero de estos oponentes: un vampiro conocido como Drácula Man, en una pelea cargada de situaciones absurdas. La historia continúa en el capítulo 73, donde Upa y Puar intervienen con transformaciones ingeniosas para ayudar a

derrotar al vampiro, demostrando que la creatividad es tan importante como la fuerza, por lo que en el capítulo 74, tiene lugar una de las batallas más memorables que es la lucha contra el Hombre Invisible, dado que el enemigo no puede verse, Krilin recurre a una estrategia poco convencional ya que logra que el Maestro Roshi sangre por la nariz al ver los senos de Bulma, revelando así la silueta del oponente con el chorro de sangre. Ya en el capítulo 75, Gokú se enfrenta a un demonio conocido como El Diablo, un enemigo que, a pesar de su apariencia intimidante, es vencido con relativa facilidad. Finalmente, en el capítulo 76, aparece el último guerrero, un misterioso Hombre Enmascarado que demuestra ser un oponente muy formidable, sin embargo, tras una intensa y emocional pelea, Gokú descubre que en realidad se trata de Son Gohan, su propio abuelo fallecido, quien ha regresado solo por un momento para enfrentarlo y probar su crecimiento.

Esta saga es ideal para un videojuego debido a la variedad de enemigos, escenarios misteriosos, retos únicos y el equilibrio entre acción, humor y emoción. En este proyecto, se tomarán como base dos de los combates más emblemáticos: el enfrentamiento contra El Hombre Invisible, que plantea un reto perceptual para el jugador, y el duelo contra Son Gohan, que mezcla acción estratégica con una narrativa emotiva. Ambos casos se traducen fácilmente en niveles diferenciados y programables dentro de las capacidades del entorno Qt.

3. Niveles del juego

3.1. Nivel 1 – El Hombre Invisible:

Este nivel se desarrolla en un escenario de desplazamiento lateral, dentro de una cámara oscura del palacio de Uranai Baba, con un ambiente misterioso y poca visibilidad. El jugador

controla a Gokú, quien puede moverse lateralmente y realizar saltos, mientras enfrenta a un enemigo que se mantiene completamente invisible durante la mayor parte del combate. A pesar de no poder verlo directamente, la presencia del Hombre Invisible se revela mediante pistas visuales y auditivas, como partículas de polvo en el aire, objetos que se mueven levemente o efectos de sonido que se activan en momentos clave. El reto principal consiste en detectar correctamente estas señales y atacar con precisión en el momento adecuado, ya que el enemigo solo puede ser dañado durante breves ventanas de vulnerabilidad.

Para lograr esta dinámica, se implementan varias físicas programadas en Qt: En primer lugar, el movimiento de Gokú incluye desplazamiento lateral y un salto parabólico simulado. Este se controla mediante eventos de teclado (keyPressEvent) y un temporizador (QTimer) que ajusta progresivamente la coordenada vertical (y()), generando una curva que emula el efecto de la gravedad. En segundo lugar, la detección de colisiones está condicionada a momentos específicos. El enemigo solo puede ser golpeado cuando su silueta aparece brevemente (aproximadamente 3 segundos), ya sea a través de un sprite de partículas o una animación de sangre que lo delate visualmente. Solo durante ese instante, el método collidesWithItem() activa el daño. Por último, el movimiento del enemigo se mantiene oculto, pero cada cierto segundo se genera una señal acústica mediante QSoundEffect y una partícula en pantalla que indica su posición aproximada. Estas señales funcionan como indicadores estratégicos de ataque o advertencia, incentivando al jugador a estar atento y actuar con rapidez.

En conjunto, este nivel propone que el jugador debe combinar observación, tiempo de reacción y control del movimiento para vencer a un enemigo invisible que solo se deja ver por breves instantes.

Objetivo nivel 1: Acertar al menos 5 golpes en el enemigo antes de que se agote el tiempo o las vidas de Gokú. Si el jugador falla demasiados ataques o es golpeado, pierde.

3.2. Ecuaciones nivel 1:

- **Movimiento parabólico del salto de Gokú:**

$$y(t) = y_0 + v_0 t - \frac{1}{2} g t^2$$

$y(t)$: Posición vertical en el tiempo

y_0 = Altura inicial

v_0 = Velocidad inicial del salto

g = Gravedad $\left(9.8 \frac{m}{s^2}\right)$ o ajustable

t = Tiempo controlado por QTimer

- **Detección condicional por colisión:**

$$\text{Daño} = \begin{cases} 1, & \text{si } t_{\text{actual}} \in [t_{\text{visible}}, t_{\text{visible}} + \Delta t] \\ 0, & \text{otro caso} \end{cases}$$

Nota: Gokú solo puede hacer daño si el enemigo está visible (Por partícula) durante Δt segundos.

- **Activación aleatoria del enemigo (Posición y sonido):** Puede ser valores aleatorios o patrón sinusoidal para agregar rigurosidad.

$$x_e = x_0 + A \sin(\omega t + \phi)$$

x_e = Movimiento oscilatorio simple en eje horizontal (enemigo inv).

A = Amplitud del movimiento

ω = frecuencia angular

ϕ = Fase

3.3. Nivel 2: El hombre enmascarado (Son Gohan):

Este segundo nivel se desarrolla en una plataforma flotante suspendida en el cielo, con vista superior al estilo de una arena mágica, con bordes peligrosos, los cuales representan una amenaza constante de caída para el jugador.

En esta etapa, Gokú se enfrenta al Hombre Enmascarado, un oponente ágil y evasivo, que más adelante se revelará como su propio abuelo, Son Gohan. El combate es más técnico que en el nivel anterior, ya que ambos personajes se mueven libremente por toda la superficie de la plataforma, intercambiando ataques y desplazamientos en todas

direcciones. Durante el duelo, si los personajes colisionan mientras están atacando, se produce una reacción de empuje que puede desestabilizar a Gokú y hacerlo caer al vacío.

Para implementar esta dinámica de forma realista en Qt, se utilizan distintas físicas y comportamientos: En primer lugar, Gokú se desplaza con movimiento libre en ocho direcciones, controlado mediante las flechas. La posición se actualiza directamente en los ejes x e y, y se puede interpolar con suavidad utilizando QTimer junto con incrementos graduales (delta) que permiten transiciones más fluidas. Además, se incorpora un efecto de deslizamiento mágico, simulando una plataforma con fricción reducida. Esto significa que, al soltar una tecla, el personaje no se detiene de inmediato, sino que continúa desplazándose unos píxeles más antes de frenar por completo. Esta inercia se controla manteniendo el movimiento activo durante algunos ciclos de QTimer, lo cual añade una capa de dificultad y precisión al control. Otro aspecto importante del combate es el empuje por colisión energética. Si Gokú y el Hombre Enmascarado chocan entre sí mientras ejecutan ataques, ambos son lanzados hacia direcciones opuestas por algunos cuadros de animación. Esta reacción se simula mediante pequeños saltos abruptos en sus coordenadas x/y, implementados y finalmente, la plataforma tiene límites claramente definidos por lo que si Gokú es empujado más allá de estos bordes durante un empuje o deslizamiento mal calculado, se activa un evento de caída, lo que le hace perder una vida automáticamente. Esto añade un componente de riesgo constante que obliga al jugador a mantener una estrategia defensiva y ofensiva bien balanceada.

Objetivo nivel 2: Reducir la barra de vida del enemigo a cero sin caer del escenario. Una vez cumplida esta condición, se activa una escena de desenlace donde el Hombre Enmascarado revela su identidad: el abuelo de Gokú, dando cierre a un combate cargado de emoción y simbolismo.

3.4. Ecuaciones nivel 2:

- **Movimiento con deslizamiento (Inercia):**

$$\vec{v}(t) \begin{cases} \vec{v}_0, & \text{mientras tecla este presionada} \\ \vec{v}_0 \cdot e^{-kt}, & \text{después de soltar la tecla} \end{cases}$$

k : Constante de desaceleración
Controlado en ciclos de QTimer

- **Detección de caída por colisión con borde:**

$$Daño = \begin{cases} 1, & \text{si } (x,y) \text{ no es valido en el area} \\ 0, & \text{otro caso} \end{cases}$$

- **Empuje tras colisión energética:**

Se traslada al jugador/enemigo unos píxeles en sentido opuesto al contacto.

Se puede usar un vector de dirección y una duración controlada por tiempo (Qtimer).

$$\Delta x = v_e \cdot \Delta t$$

$$v_e = \text{Velocidad de empuje}$$