

MOMENTO II

Autores: *Maria Valentina Quiroga Alzate, Juan Felipe Orozco Londoño*

Informática II

*Departamento de Ingeniería Electrónica y de Telecomunicaciones
Universidad de Antioquia*

0.1. Capítulos escogidos

Para el desarrollo del videojuego se seleccionaron los episodios comprendidos entre el número 70 y el 76 de la serie *Dragon Ball*, correspondientes a la saga del Palacio de Uranai Baba. Esta elección se fundamenta en la riqueza de combates creativos, así como en la variedad de personajes y escenarios, los cuales resultan altamente adaptables al diseño de niveles para un videojuego sencillo. Además, las características de esta saga permiten implementar físicas jugables en Qt de manera coherente con la lógica de juego y con posibilidades técnicas accesibles.

0.2. Detalles adicionales y cambios respecto al Momento I

0.2.1. Confirmación y diferenciación de modelos físicos

En la entrega del Momento I se propusieron tres modelos físicos para representar fenómenos relevantes dentro del juego. En esta sección se confirma su implementación, se especifica su rol en la experiencia de usuario y se describe su aplicación desde el punto de vista técnico.

El primer modelo corresponde al **movimiento parabólico**, utilizado para representar el salto de Gokú en el primer nivel. Este comporta-

miento está basado en una ecuación cuadrática dependiente del tiempo, que simula la gravedad sobre un cuerpo proyectado. Su implementación, a través de la clase *FisicaSaltoParabolico*, permite ajustar parámetros como la velocidad inicial, la gravedad y la duración del salto. Esta elección enriquece la jugabilidad al permitir maniobras evasivas, establecer zonas de riesgo y controlar la dificultad del nivel mediante la trayectoria del salto.

El segundo modelo se refiere al **deslizamiento con fricción reducida**, empleado en el segundo nivel, donde el jugador se desplaza sobre una plataforma flotante. En este caso, cuando el jugador deja de presionar las teclas de dirección, Gokú no se detiene de inmediato, sino que continúa su movimiento con una inercia decreciente. Esta dinámica está representada mediante la clase *FisicaDeslizamiento*, la cual aplica una deceleración exponencial del tipo $v(t) = v_0 \cdot e^{-kt}$. Dicho comportamiento introduce un desafío adicional, pues exige precisión en el control y anticipación de movimientos, incrementando el riesgo de caída si el jugador no calcula bien su trayectoria.

El tercer modelo consiste en el **empuje energético por colisión**, utilizado cuando Gokú y el enemigo chocan en medio de un ataque. Esta física, implementada en la clase *FisicaEmpujeEnergetico*, consiste en aplicar una traslación rápida sobre las coordenadas del personaje durante un corto intervalo, generando un retroceso que modifica la posición de ambos personajes. Esta reacción no solo genera impacto

visual inmediato, sino que también tiene consecuencias estratégicas, como el riesgo de ser empujado al borde y caer fuera del área segura.

0.2.2. Precisión en los tiempos y condiciones de visibilidad

Para que la mecánica del primer enemigo, el Hombre Invisible, sea jugable y coherente, se definieron con mayor rigor los tiempos de visibilidad y las condiciones asociadas a las colisiones. El enemigo será visible durante un intervalo de tres segundos, tiempo suficiente para que el jugador detecte su presencia mediante efectos visuales o sonoros y pueda ejecutar un ataque básico. Esta duración fue determinada considerando que el jugador necesita alrededor de un segundo para percibir la aparición y al menos dos segundos para reaccionar.

Entre cada aparición, el enemigo permanecerá oculto durante un periodo aleatorio de entre cuatro y seis segundos, definido mediante un generador aleatorio y controlado por un `QTimer`. De esta forma, se evita que el jugador memorice el patrón de visibilidad, obligándolo a mantener una atención sostenida. Durante los periodos en que el enemigo es visible, se activa su zona de colisión, permitiendo que Gokú cause daño si su ataque coincide con dicha ventana.

Desde el punto de vista técnico, esta lógica es fácilmente implementable en Qt mediante las clases `QTimer`, `QEapsedTimer` y `QSoundEffect`. Las colisiones se manejan con el método `collidesWithItem()` entre objetos derivados de `QGraphicsItem`. Además, los parámetros de visibilidad pueden ser ajustados para modificar la dificultad en distintos niveles o modos de juego.

0.2.3. Confirmación del modelo de colisión por área

Para garantizar la precisión de las interacciones entre personajes y entorno, se confirma el uso del modelo de colisión basado en la superposición de áreas. Este modelo se aplica tanto en situaciones ofensivas (como ataques y empujes) como en situaciones defensivas (como detección de bordes o caídas). Su implementación se realiza con el método `collidesWithItem()` de la clase `QGraphicsItem`, propio del entorno Qt.

En el primer nivel, la zona de colisión del Hombre Invisible solo se activa durante los períodos de visibilidad. La figura geométrica que representa esta zona podrá ser un rectángulo que encapsule la silueta parcial del sprite. Cuando Gokú ejecuta un ataque durante este intervalo, se evalúa si su área se superpone con la del enemigo y, en caso afirmativo, se considera un impacto válido.

En el segundo nivel, las colisiones cumplen una doble función. Por un lado, al chocar ambos personajes mientras están en estado de ataque, se activa una reacción de empuje que desplaza a ambos en direcciones opuestas. Este empuje se simula mediante un desplazamiento controlado en las coordenadas x o y. Por otro lado, tras el empuje, se verifica si Gokú se encuentra fuera del área válida de la plataforma (definida como un `QRectF`). En tal caso, se activa un evento de caída.

0.2.4. Clarificación de objetivos por nivel

En el documento del Momento I se establecieron los objetivos generales de cada nivel. En esta sección se detallan con mayor precisión los criterios de éxito, las condiciones de derrota y los mecanismos de seguimiento que serán utilizados para validar dichos criterios en tiempo de ejecución, principalmente mediante la clase `Nivel` y

su método `estaCompletado()`.

En el **Nivel 1**, el jugador debe acertar al menos cinco golpes válidos al Hombre Invisible antes de que se agote el tiempo límite o se queden sin vidas. El cumplimiento de este objetivo requiere que los golpes se efectúen durante las ventanas de visibilidad del enemigo, y que Gokú conserve al menos una vida cuando se alcanza el quinto golpe. El sistema controla este progreso mediante variables como `golpesAcertados`, `vidas` y un temporizador global implementado con `QEapsedTimer`. En caso de éxito, se reproduce una animación de retroceso del enemigo, se oscurece el escenario y se presenta un mensaje de victoria.

En el **Nivel 2**, el objetivo es reducir completamente la barra de vida del Hombre Enmascarado sin que Gokú caiga fuera de la plataforma. La condición de éxito implica que la variable `vidaEnemigo` llegue a cero y que Gokú se mantenga dentro del área válida. La derrota ocurre si Gokú es empujado fuera por una colisión o si cae por error de cálculo. Las métricas de seguimiento incluyen el estado de la vida del enemigo, la posición de Gokú y una posible bandera `estaCayendo` que puede utilizarse para activar una animación de derrota. En caso de victoria, se reproduce una escena narrativa donde Gokú reconoce al enemigo como su abuelo, cambia la música y se finaliza el juego con un mensaje de cierre y opción de volver al menú principal.

0.3. Descripción detallada de vistas e interacciones

Nivel 1 – El Hombre Invisible

Este nivel se desarrolla en un escenario con vista lateral desplazable horizontalmente, ambientado en un pasillo oscuro dentro del palacio de Uranai Baba. La perspectiva general se asemeja a un *side-scroller*, donde el jugador se

desplaza de izquierda a derecha sobre una superficie continua.

El entorno cuenta con zonas clave que incluyen el piso principal —que representa el recorrido habitual de Gokú— y varios puntos donde el enemigo invisible puede aparecer. Estos puntos pueden estar definidos en posiciones fijas o generarse aleatoriamente para aumentar la incertidumbre. En la parte superior de la pantalla se ubica la interfaz de usuario (UI), que muestra el tiempo restante, una barra de vida para Gokú y un contador de golpes acertados al enemigo.

La ambientación incluye objetos móviles como jarrones que caen, partículas de polvo o cortinas en movimiento. Estos elementos no solo enriquecen visualmente el entorno, sino que también actúan como pistas indirectas para inferir la presencia del enemigo invisible.

En cuanto a los elementos visibles, el jugador controla a Gokú, quien es completamente visible durante todo el nivel. El enemigo, por su parte, permanece oculto la mayor parte del tiempo, pero aparece brevemente en forma de silueta o sprite parcial. También se visualizan efectos como partículas (polvo, humo, sangre), un indicador de sonido, una barra de vida y el contador de impactos al enemigo (de 0 a 5), así como un reloj regresivo que marca el tiempo restante para completar el objetivo.

Las interacciones fundamentales en este nivel incluyen el desplazamiento horizontal y el salto de Gokú, así como la posibilidad de atacar. Si el jugador ejecuta un ataque mientras el enemigo está visible y ambos colisionan, se contabiliza un golpe válido. En cambio, si el enemigo ataca a Gokú, este pierde una vida y se activa una animación de retroceso.

La física aplicada en este nivel incluye el modelo de salto parabólico, que se activa al presionar la tecla de salto (barra espaciadora). Cuando el enemigo es visible, su área de colisión se activa, permitiendo que el ataque de Gokú tenga efecto si coincide con su posición. Además, si

Gokú recibe daño, se ejecuta una animación de retroceso simulando un golpe recibido.

Los controles disponibles en este nivel son simples e intuitivos: las teclas de flecha izquierda y derecha permiten mover a Gokú, la barra espaciadora activa el salto, y la tecla X ejecuta el ataque básico con detección de colisión activa.



Figura 1: Primer nivel.

Nivel 2 – El Hombre Enmascarado (Son Gohan)

El segundo nivel presenta una perspectiva completamente distinta, con una vista superior (*top-down*) sobre una plataforma flotante en el cielo. El entorno tiene forma circular y está rodeado por el vacío, representado mediante un fondo de cielo, nubes o estrellas. Los personajes inician en el centro de la plataforma, y los bordes constituyen una zona de riesgo: cualquier jugador que salga de este límite será derrotado.

La interfaz superior del nivel muestra la vida de Gokú y la del enemigo, además de indicadores de estado relevantes para el combate. Entre los elementos visibles se encuentran Gokú, el Hombre Enmascarado (Son Gohan), los efectos visuales de energía durante los empujes o colisiones, la plataforma delimitada y las respectivas barras de vida. Al finalizar el nivel, se muestra un

mensaje de victoria o derrota, dependiendo del resultado.

Las interacciones permiten que ambos personajes se muevan libremente en ocho direcciones dentro del área. Cuando ambos atacan y colisionan simultáneamente, se genera un empuje que los desplaza en direcciones opuestas. Si Gokú cae por el borde de la plataforma, pierde automáticamente; por el contrario, si logra reducir completamente la vida del enemigo, el jugador gana el nivel.

En cuanto a la física aplicada, el movimiento libre se permite mediante las teclas de flechas o el conjunto WASD. Cuando el jugador suelta la tecla direccional, Gokú no se detiene de inmediato, sino que continúa desplazándose unos píxeles más, simulando una fricción reducida mediante la física de deslizamiento mágico. El empuje energético se activa si ambos personajes colisionan durante un ataque, provocando un desplazamiento temporal. Adicionalmente, si la posición de Gokú sale de los límites de la plataforma definidos mediante un objeto `QRectF` se activa una condición de derrota.



Figura 2: Segundo nivel.

0.4. Sprites utilizados por nivel

En cumplimiento con los requisitos establecidos para el diseño del videojuego, se han definido y utilizado al menos cuatro sprites distintos por nivel. Estos sprites representan personajes, enemigos, elementos del entorno y efectos visuales relevantes para la jugabilidad.

En el Nivel 1, los sprites incluyen: Gokú (jugador), el enemigo invisible en su forma parcialmente visible, objetos interactivos como jarrones o cortinas, y efectos visuales como polvo o humo. Para el Nivel 2, se incorporan los sprites de Gokú, el Hombre Enmascarado, la plataforma flotante y los efectos de energía generados durante las colisiones.



Figura 3: Sprite Bulma.

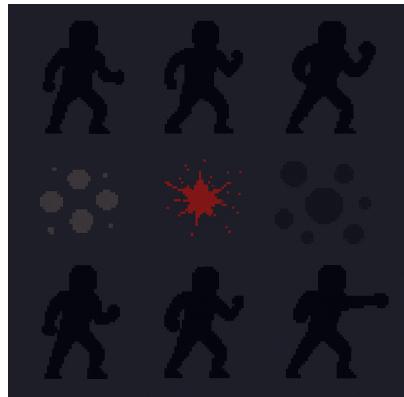


Figura 4: Sprite hombre invisible.



Figura 5: Sprite primer nivel.



Figura 6: Sprite segundo nivel.

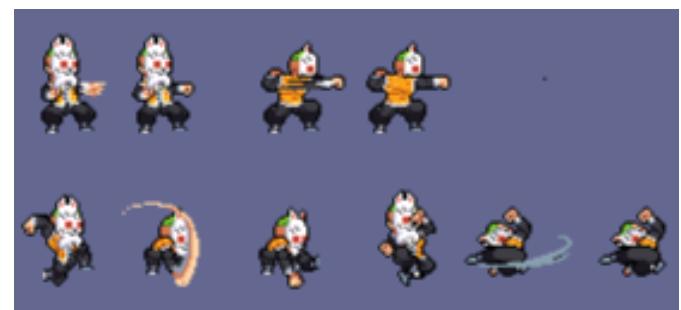


Figura 7: Sprite tercero nivel.

El uso de estos sprites permite representar de manera coherente las acciones, eventos y condiciones del juego, enriqueciendo la experiencia visual sin comprometer el rendimiento de la aplicación.

0.5. Diagrama de clases

A continuación se presenta el diagrama de clases correspondiente a la capa lógica del sistema. En él se representan las clases que intervienen en la simulación del comportamiento físico, la interacción entre personajes, los niveles del juego y la validación de condiciones de éxito o derrota.

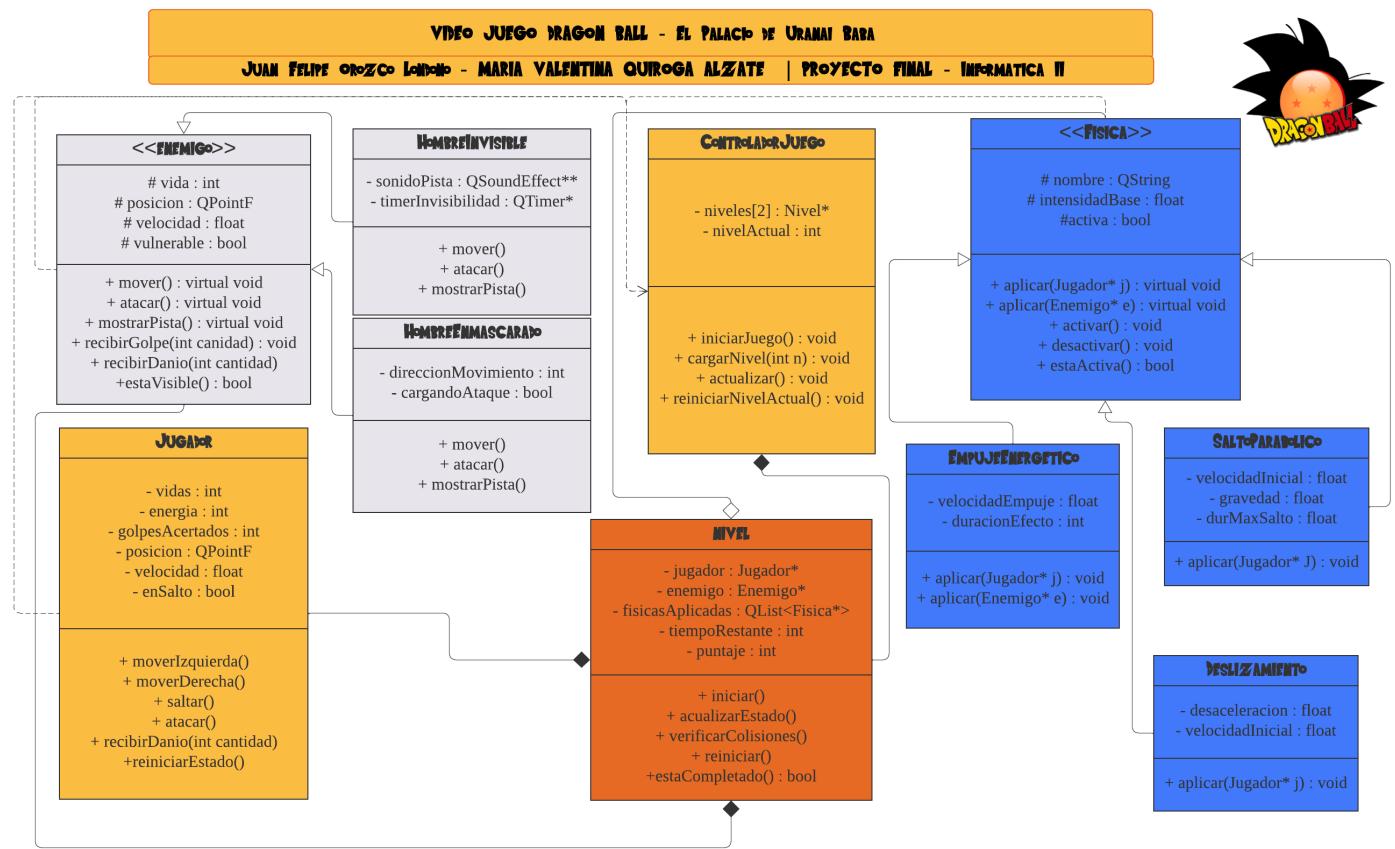


Figura 8: Diagrama de clases.

Bibliografía

Kanzenshuu, *Dragon Ball Episode 076*, Kanzenshuu, [Online]. Available: <https://www.kanzenshuu.com/episode/db/eps-076/>.

Pierre2011amkpopo, *Grandpa Gohan sprite sheet – Dragon Ball classic*, DeviantArt, [Online]. Available: <https://www.deviantart.com/pierre2011amkpopo/art/Grandpa-gohan-sprite-sheet-Dragon-ball-classic-1177609805>.