Proyecto Final

Samuel Zuleta Zapata

Juan Fernando Henao Ledesma

Informática II

Aníbal Guerra Soler y Augusto Salazar

Universidad de Antioquia

Medellín

# Capítulo 53 – “¡Goku contra el General Blue!”

Saga del Ejército Red Ribbon, Saga #3

## Contextualización:

En el capítulo 53 de Dragon Ball, Goku continúa su búsqueda de las esferas del dragón dentro de una antigua base submarina conocida como la Cueva del Pirata. Junto a Bulma y Krilin, se enfrenta a uno de los miembros más peligrosos del Ejército Red Ribbon: el General Blue, un enemigo que destaca por su inteligencia, su carácter calculador y una habilidad especial: puede paralizar a sus víctimas con la mirada gracias a sus poderes psíquicos.  
  
En este episodio, Goku enfrenta un nuevo tipo de amenaza: trampas mecánicas, enemigos artificiales y un ambiente hostil que pone a prueba su ingenio, agilidad y fuerza. El episodio mezcla tensión, acción y humor, y representa una evolución en la narrativa de Dragon Ball, al incluir enemigos que no solo son físicamente fuertes, sino que también presentan desafíos mentales y tecnológicos.

## Ideas del desarrollo del juego:

**Nivel #1: *Resistencia en la Cueva del Pirata***

En este nivel, el jugador controla a Goku dentro de la base submarina mientras esta comienza a colapsar lentamente. El objetivo es resistir durante 2 minutos dentro del área, esquivando peligros y sobreviviendo a una oleada de obstáculos constantes.

* Objetivo: mantenerse con vida durante 2 minutos esquivando ataques y trampas sin salir del área.
* Obstáculos: escombros que caen desde el techo
* Mecánicas: salto y esquive
* Ambiente: la cueva del pirata, sonido de alarmas y ambiente oscuro
* Sistema de vida: se utiliza una barra de vida que inicia con 100 puntos. Cada impacto con obstáculos resta una cantidad especifica de vida. Si la barra llega a 0 antes de que acabe el tiempo, el jugador pierde.

Si el jugador logra aguantar los 2 minutos sin perder todas sus vidas, pasa al siguiente nivel.

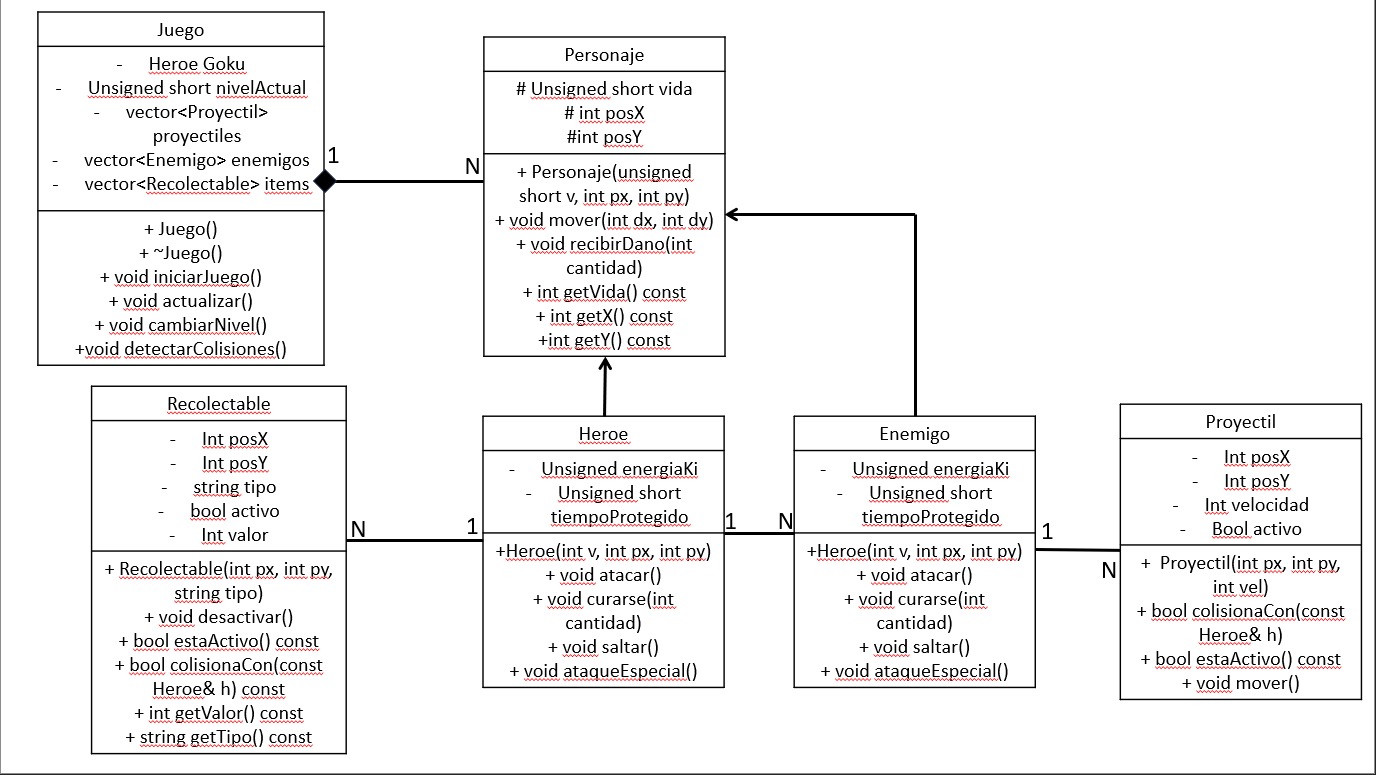
### Nivel #2: Enfrentamiento contra los Robots del General Blue

En este nivel, el jugador no lucha directamente contra el General Blue, sino contra un grupo de 20 robots de combate activados por él como defensa final.

* Objetivo: derrotar a los 20 robots antes de que se agote el tiempo (2 minutos).
* Mecánicas: combate cuerpo a cuerpo
* Sistema de vida: se utiliza una barra de vida para Goku que inicia en 100 puntos. Cada ataque recibido por parte de los robots reduce esta barra. Si la vida llega a 0 antes de eliminar a los 20 enemigos, el jugador pierde.

Cuando el jugador derrota a los 20 robots. Se muestra una animación breve con Goku triunfante saliendo de la base submarina.

1. **Diagrama de clases**



**2. Cambios con respecto al primer avance**

Desde la entrega del Momento I, se han realizado varias modificaciones importantes al diseño del juego, que enriquecen tanto la jugabilidad como la coherencia temática con el universo Dragon Ball. Estas son las principales actualizaciones

* Cambio de los enemigos del segundo nivel: originalmente se planeaban robots del Ejercito Red Ribbon. Sin embargo, debido a la dificultad para conseguir sprites apropiados, se reemplazaron por soldados del Ejército Red Ribbon.
* Introducción de un sistema de acumulación de energía (ki): ahora, a medida que Goku derrota enemigos en el segundo nivel, su nivel de ki irá aumentando progresivamente. Esta barra de ki permite que el jugador, tras eliminar cierta cantidad de enemigos o tras cierto tiempo, pueda desatar un ataque especial como un Kamehameha que inflige un gran daño a los enemigos.
* Añadido de enemigos con ataques a distancia: ahora los enemigos están equipados con armas, lo que obliga al jugador a esquivar, agregando complejidad al combate.
* Aparición de recolectables curativos tipo “semillas del ermitaño”: durante los niveles, aparecerán de forma aleatoria unos ítems recolectables que representan semillas del ermitaño, las cuales restauran parte de la vida de Goku al ser recogidas.

**3. Descripción de vistas, interacciones y físicas del juego**

Nivel 1 – Resistencia en la Cueva del Pirata

Este nivel se desarrolla en una vista lateral 2D. El jugador controla a Goku, quien debe sobrevivir dentro de una base submarina que comienza a colapsar, esquivando rocas que caen desde el techo.

Interacciones y físicas:

* Las piedras caen en caída libre vertical, acelerándose naturalmente hacia el suelo, lo que simula la gravedad.
* Goku puede moverse horizontalmente a velocidad constante (movimiento rectilíneo uniforme), permitiendo esquivar obstáculos hacia la izquierda o derecha.
* También puede saltar, siguiendo una trayectoria curva en el aire (movimiento parabólico), lo que permite esquivar rocas que caen o alcanzar recolectables.

Nivel 2 – Enfrentamiento contra los Soldados Red Ribbon

Este nivel también se presenta en vista lateral 2D, con desplazamiento libre por el escenario. Goku se enfrenta a varios soldados del Ejército Red Ribbon en combate.

Interacciones y físicas:

* Los proyectiles enemigos no siguen una línea recta, sino que se mueven en una trayectoria ondulada o senoidal, lo que hace más difícil predecirlos y esquivarlos.
* Goku puede atacar de forma directa y también puede acumular energía (ki) al eliminar enemigos. Cuando alcanza suficiente nivel de ki, puede ejecutar un ataque especial poderoso, como un Kamehameha, que inflige daño en un área más amplia.

Momento final.

Durante la implementación se definieron mas clases de las previstas inicialmente y se eliminó una que se pensaba añadir en un principio. Las clases que componen el proyecto son la clase personaje que hereda a la clase goku y a la clase enemigo, la clase juego que sirve como mecanismo de control para organizar la interfaz gráfica, dar inicio a los niveles y controlar el flujo del juego, la clase obstáculo que hereda a la clase roca y a la clase proyectil, siendo estos objetos que aparecen en el nivel 1 y 2 respectivamente, cada uno posee mecánicas de movimiento distintas que van a ser explicadas más adelante. Respecto al diagrama de clases original se descarta la clase recolectable ya que consideramos que sería disminuir significativamente la dificultad de los niveles.

Goku se puede desplazar de izquierda a derecha con velocidad constante, teniendo también la posibilidad de saltar, simulando un movimiento parabólico y siendo uno de los fenómenos físicos implementados. Goku tras recibir algún impacto que le haga daño va a tener 1.5 segundos de invencibilidad que se van a evidenciar por un parpadeo en el Sprite.

El primer nivel consta de un escenario de cueva donde el objetivo de goku va a ser evitar las piedras que caen desde el techo simulando una caída libre, la implementación de este nivel sigue con lo planeado en los momentos iniciales, las piedras aparecen en una posición aleatoria en x, ya que en y siempre aparecen en 0. Las piedras se eliminan automáticamente cuando salen de la escena o cuando golpean al personaje, estas tienen un daño de 20 y goku debe aguantar durante 1:30 segundos para avanzar de nivel.

El segundo nivel es en un escenario de laboratorio de la red ribbon, donde el héroe se va a enfrentar a 20 soldados, estos van a aparecer en posiciones cercanas a goku y van a disparar proyectiles que se desplazan de manera senoidal, goku va a recibir daño de dos formas en este nivel, la primera es claramente si un proyectil lo impacta, la segunda forma va a ser que goku no colisione con los enemigos por la parte de arriba de estos sino lateralmente, ocasionándole en ambos casos un daño de 20; la condición de victoria para este nivel va a ser derrotar a 20 enemigos. En este nivel se descartó la idea del ataque final por cuestiones de tiempo.

Como contenedores para las rocas en el primer nivel y para los enemigos en el segundo nivel se usan las Qlist de Qt, ya que tras haber analizado y consultado sobre estos contenedores nos dimos cuenta que son los más óptimos para almacenar nuestras clases, ya que cuentan con métodos útiles de acceso, inserción y eliminación, siendo fáciles de recorrer para verificar las colisiones entre objetos.

Diagrama de clases. (Cajas con atributos y métodos más abajo)

|  |
| --- |
| QMainWindow |
|  |
|  |

|  |
| --- |
| MainWindow |
|  |
|  |

|  |
| --- |
| Juego |
|  |
|  |

|  |
| --- |
| QObject |
|  |
|  |

|  |
| --- |
| QGraphicsPixmapItem |
|  |
|  |

|  |
| --- |
| QObject |
|  |
|  |

1

1

N

|  |
| --- |
| Personaje |
|  |
|  |
|  |

|  |
| --- |
| QGraphicsPixmapItem |
|  |
|  |

|  |
| --- |
| Obstaculo |
|  |
|  |

N

|  |
| --- |
| Enemigo |
|  |
|  |

|  |
| --- |
| Goku |
|  |
|  |

|  |
| --- |
| Proyectil |
|  |
|  |

|  |
| --- |
| Roca |
|  |
|  |

N

1

|  |
| --- |
| Juego |
| * QGraphicsView\* view * QGraphicsScene\* scene * QTimer\* timer * Goku\* goku * QGraphicsRectItem\* vidaBar * QGraphicsTextItem\* notaNivel * QGraphicsTextItem\* tiempoText * QGraphicsTextItem\* sombraTiempo * QGraphicsTextItem\* textoVida * QGraphicsTextItem\* sombraVida * QGraphicsTextItem\* textoNombre * QGraphicsTextItem\* sombraNombre * int tiempoRestante * int nivelActual * bool nivelTerminado * bool ganoNivel * QList<Enemigo\*> enemigos * int enemigosEliminados * int enemigosParaGanar * QGraphicsTextItem\* textoEnemigos * QGraphicsTextItem\* sombraEnemigos |
| + Juego(QGraphicsView\* view, QObject\* parent = nullptr)  + void iniciarNivel1()  + void iniciarNivel2()  # bool eventFilter(QObject\* obj, QEvent\* event) override   * void limpiarEscena() |

|  |
| --- |
| Personaje |
| # int vida  # int velocidad  # bool inmune  # QTimer\* timerInmunidad  # QTimer\* timerParpadeo |
| + Personaje()  + void setVida(int v)  + int getVida() const  + void restarVida(int cantidad)  + void curar(int cantidad)  + void setVelocidad(int v)  + int getVelocidad() const |

|  |
| --- |
| Goku |
| * int spriteX * int spriteY * int spriteAncho * int spriteAlto * QPixmap hojaSprites1 * QPixmap hojaSprites2 * QPixmap sprite * QPixmap spriteSaltoDerecha * QPixmap spriteSaltoIzquierda * int cont * int velocidadY * bool enElAire * bool moviendoIzq * bool moviendoDer * bool mirandoDerecha |
| + Goku()  + void aplicarFisica()  + void confSprite(bool band)  # void keyPressEvent(QKeyEvent\* event) override  # void keyReleaseEvent(QKeyEvent\* event) override   * bool esOpaco() const |

|  |
| --- |
| Enemigo |
| * QTimer\* timerDisparo * bool mirandoIzquierda |
| + Enemigo(qreal posGoku)  + void disparar() |

|  |
| --- |
| Obstaculo |
| # int velocidad |
| + Obstaculo()  + virtual void mover() |

|  |
| --- |
| Roca |
|  |
| + Roca()  + void mover() override |

|  |
| --- |
| Proyectil |
| * int sprite * int spriteY * int spriteAncho * int spriteAlto * QPixmap hojaSprites * QPixmap sprite * int cont * bool direccionIzquierda * QTimer\* timerMovimiento * int velocidadX * double amplitud * double frecuencia * qreal x\_inicial * qreal y\_inicial |
| + Proyectil(bool haciaIzquierda, qreal x\_origen, qreal y\_origen)  + void mover() override  + void confSprite() |

|  |
| --- |
| MainWindow |
| * QGraphicsView\* view * Juego\* juego * QWidget\* menuWidget * QPushButton\* btnIniciar * QPushButton\* btnSalir |
| + MainWindow(QWidget \*parent = nullptr)  + ~MainWindow()   * void iniciarJuego() * void salir() |