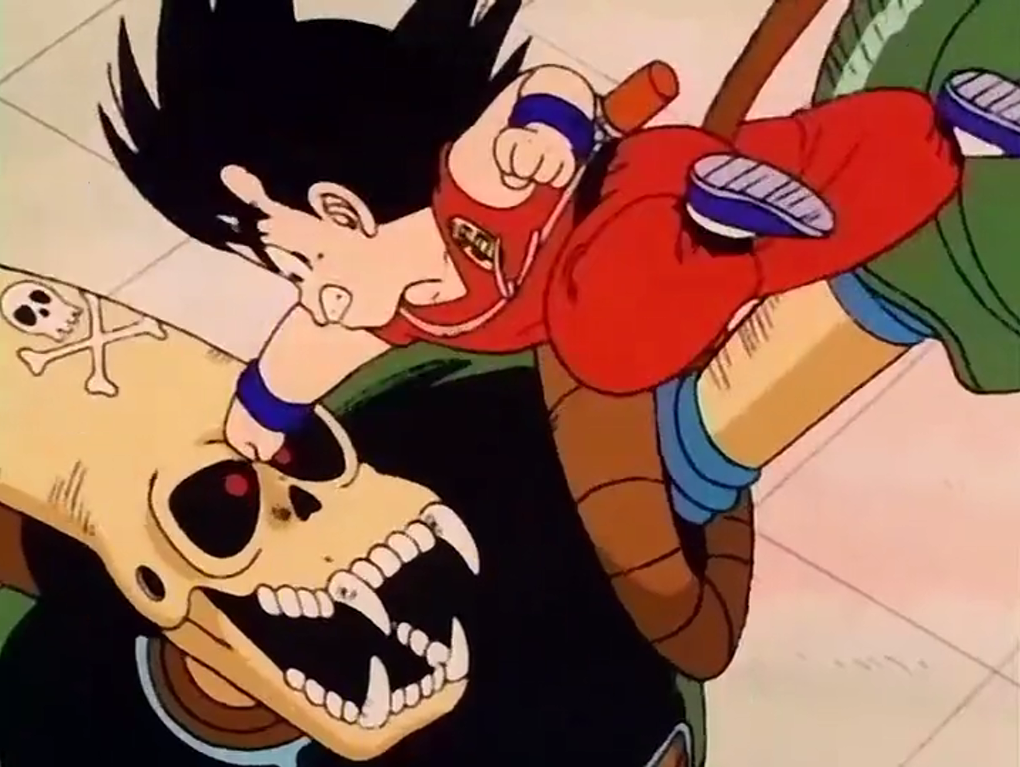
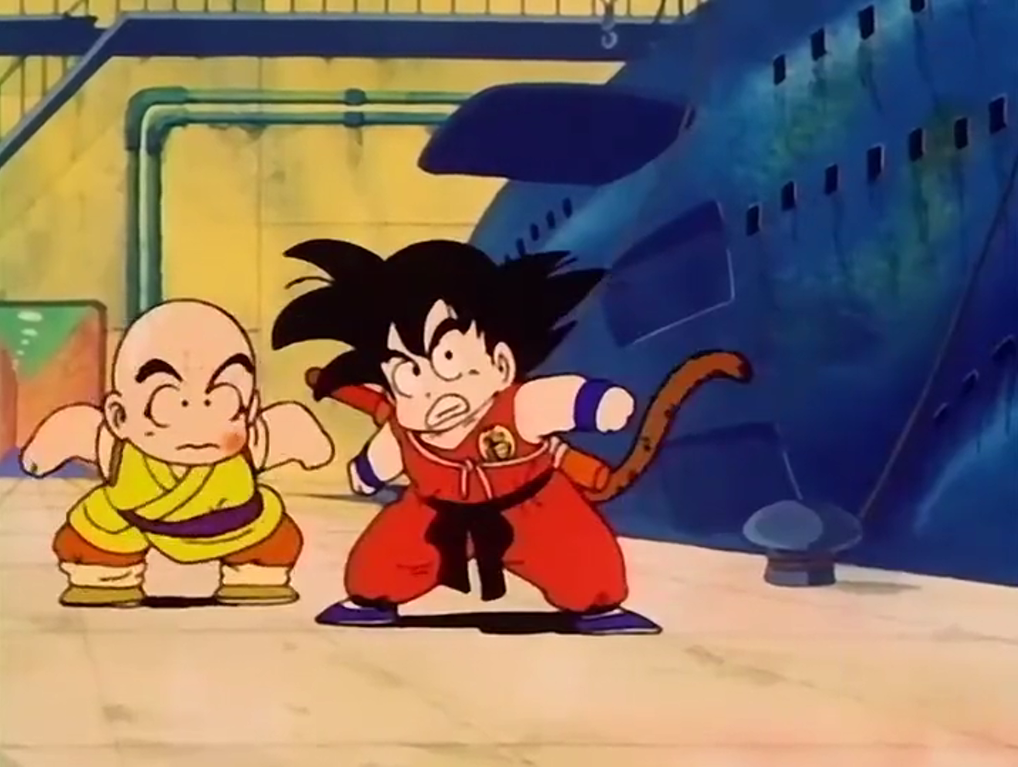
**Informe Inicial Proyecto Final - Informatica II (2025-1, UdeA)**

Integrante: Jose Eduardo Valverde Álvarez

**Contextualización**

**Capitulos Seleccionados: #51 - El guardián del fondo del mar**

**Sinopsis del capítulo:**

Gokú y Krilin se enfrentan al robot guardián, pero tienen muchas dificultades para acabar con él: Ni el cañón de un barco ni arrollarlo con un camión parecen poder derrotarle, así que todos comienzan a huir. No es hasta que Gokú, desde lo alto de un edificio, logra destruirle la cabeza de un fuerte puñetazo en caída libre, pero provocando inadvertidamente el colapso de la gruta.

**Físicas/Mecánica del juego**

* **Objetivo y Movimientos (Nivel 1 y 2)**

El personaje combatiente (Goku) puede moverse de cualquier (caminar, correr, e incluso saltar) con tal de atacar físicamente al enemigo que el robot pirata, por cada interacción de ataque que el personaje va creando al enemigo puede acumular puntos para desbloquear un tipo de ataque definitivo que permite derrotar al enemigo tal y como se muestra en el capítulo original.

El enemigo se moverá constantemente con tal de atacar físicamente a Gokú, solo de forma básica, pero no podrá ser derrotado con el fin de usar el ataque definitivo tal como en el capítulo, pero acá solo se utilizará algo simple como un salto parabólico.

* **Fisicas y otras mecánicas (en ambos niveles):**
  + Movimiento parabólico (en salto y movimiento definitivo)
  + Choque o colisión (interacción del jugador con obstáculos)
  + Knockback (al chocar o recibir un ataque del enemigo el personaje hace un movimiento de retroceso se pierde una vida, y viceversa)
  + Estado defensivo (solo por parte de Gokú)

**Retos**

Para el primer nivel, Gokú debe acumular los ataques para acumular puntos hasta llenar la barra en donde se desbloquea el ataque definitivo, que es el único en donde se puede derrotar completamente al enemigo.

En el segundo nivel, el objetivo será el mismo pero se requerirá mas habilidad para momentos como las caídas de los obstáculos (piedras), en el escenario

**Momento II – Diagrama y Descripción de las clases**

**Diagrama UML:**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Detalles del Diagrama**

**Clase juego**

**Propósito:** Controlador principal del videojuego

* **Atributos:** 
  + niveles: Cantidad total de niveles en el juego
  + start: Estado de inicio del juego
  + end: Estado de finalización del juego
* **Métodos:** 
  + iniciarJuego(): Inicia una nueva partida
  + finalizarJuego(): Termina la partida actual
  + cambioNivel(): Gestiona la transición entre niveles

**Clase nivel**

**Propósito:** Representa cada etapa individual del juego

* **Atributos:** 
  + nombreNivel: Identificador del nivel
  + escenario: Imagen de fondo del nivel
  + ancho y alto: Dimensiones del área de juego
  + Referencias a personajes: gokuJugador, robotEnemigo
* **Métodos:** 
  + actualizarPartida(): Actualiza el estado del nivel en cada frame
  + nivelFinalizado(): Verifica si se completaron los objetivos del nivel

**Clase Personaje (Clase Base)**

**Propósito:** Define las características comunes de todas las entidades del juego

* **Atributos Comunes:** 
  + x, y: Posición en coordenadas 2D
  + vidas: Puntos de vida del personaje (tendrá 3 asignados)
  + activo: Estado de activación en el juego
* **Métodos Base:** 
  + mover(): Movimiento básico
  + estaVivo(): Verificación de estado del personaje (si fue derrotado o no)

**Clase Goku (Hereda de Personaje)**

**Propósito:** Personaje protagonista controlado por el jugador

* **Atributos Especiales:** 
  + definitivaGauge: Medidor de energía para el ataque definitivo
  + definitivaDisponible: Disponibilidad del ataque definitivo
* **Habilidades de Movimiento:** 
  + moverIzquierda(), moverDerecha(): Desplazamiento horizontal
  + salto(): Movimiento vertical
* **Habilidades de Combate:** 
  + golpeBasico(): Ataque cuerpo a cuerpo
  + ataqueDefinitivo(): Ataque especial de alto daño, suficiente para derrotar al robot
* **Sistema de Energía:** 
  + recargarDefinitivaGauge(): Acumula energía especial
  + recibirDaño(): Procesa daño recibido por el enemigo (en este caso -1 punto de vida)

**Clase Robot (Hereda de Personaje)**

**Propósito:** Enemigos automatizados del juego

* **Atributos Específicos:** 
  + cooldown: Tiempo de espera entre ataques
* **Comportamientos del robot:** 
  + recibirAtaqueBasico(): Respuesta a ataques del jugador
  + recibirAtaqueDefinitivo(): Respuesta a ataques especiales
  + choqueGoku(): Interacción de colisión con el protagonista

**Clase Piedras (Hereda de Personaje)**

**Propósito:** Obstáculos ambientales interactivos

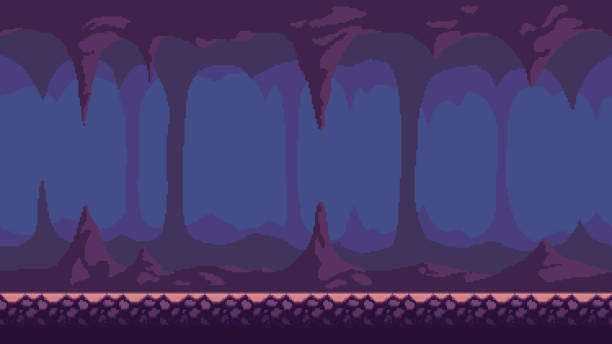
* **Atributos:** 
  + activa: Estado de activación del obstáculo
* **Mecánicas en el escenario:** 
  + lluvia(): Genera lluvia de piedras como peligro
  + activarCaida(): Activa la caída de piedras cuando el jugador se acerca

**Relaciones entre Clases**

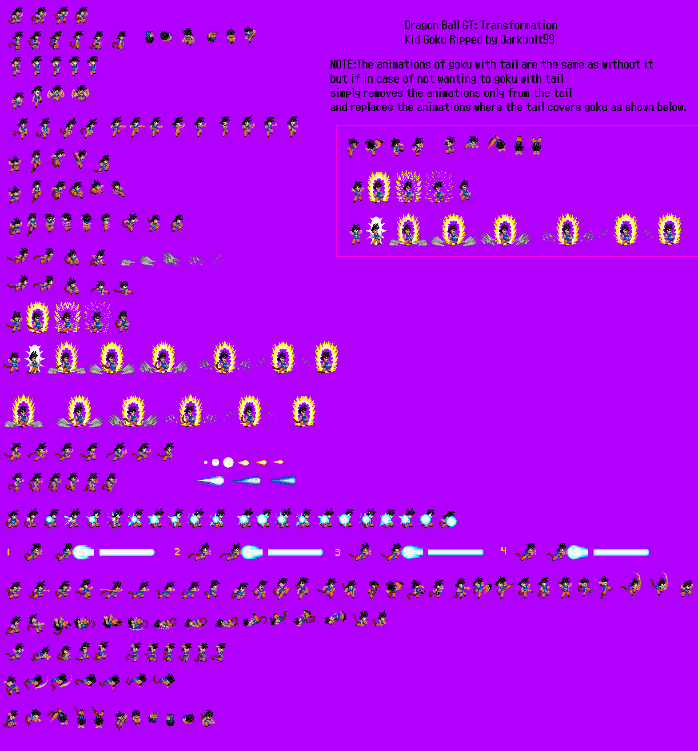
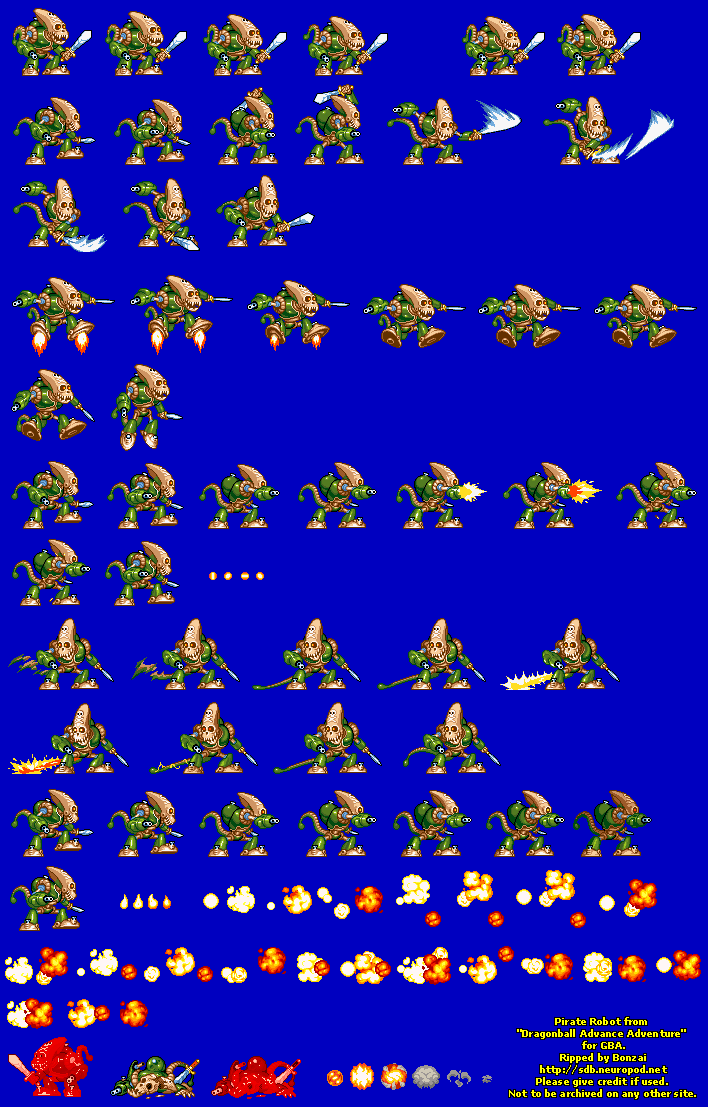
* Composición: juego contiene múltiples niveles
* Agregación: Cada nivel referencia instancias de Gokú, Robot y Piedras
* Herencia: Gokú y Robot son clases con atributos únicos adicionales que identifican al tipo de personaje en el juego
* Interacción: Los personajes interactúan entre sí mediante métodos de ataque, colisión y daño

**(Posibles) Sprites**

**Fondo**



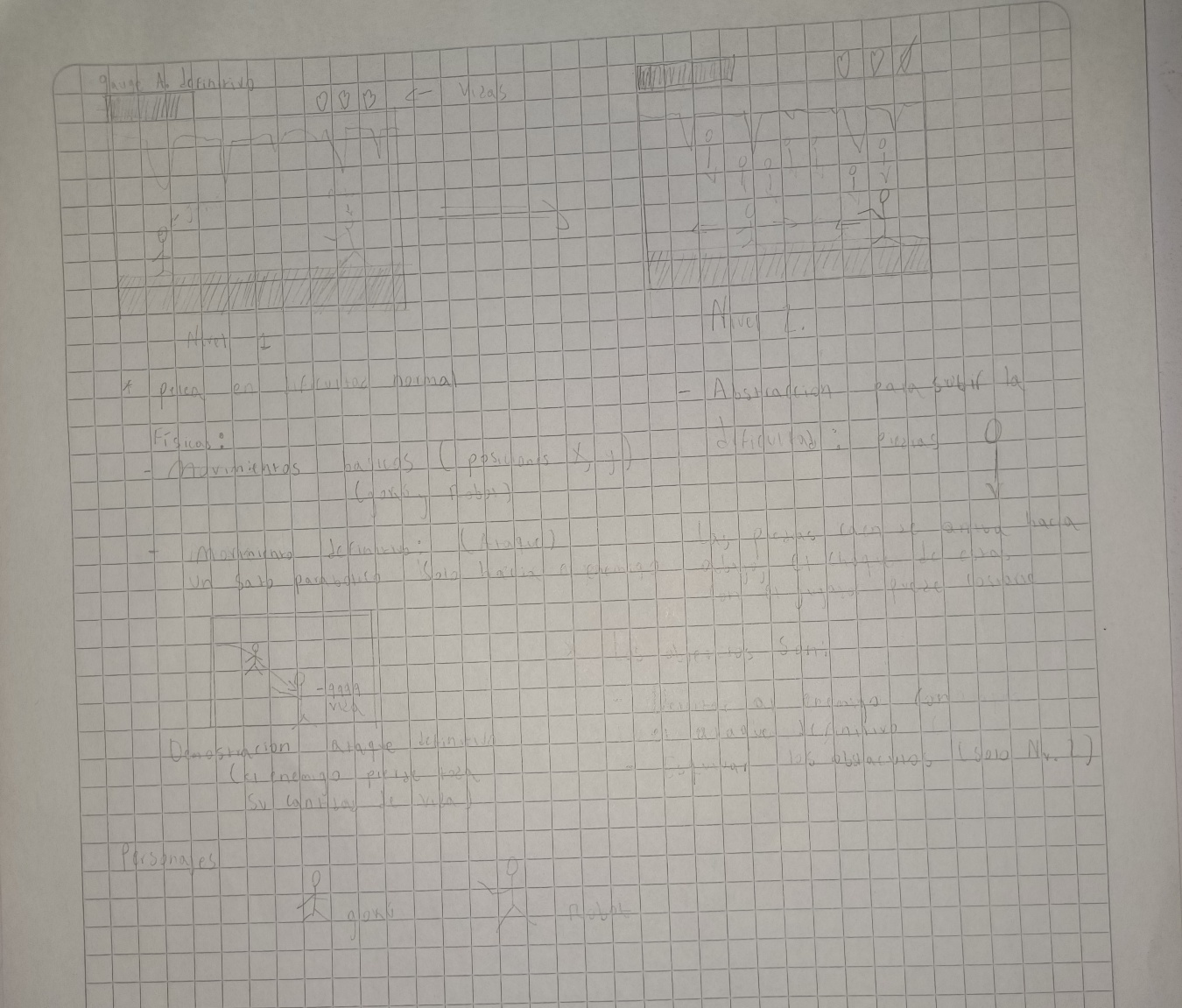
**Personajes**

**** 

**Obstaculo:**



**Ilustración gráfica del juego**

****