Momento III

Momento I:

**Momento I**

El juego que crearemos será de tipo lucha y acción, en el cual Goku participará en el Torneo Mundial de Artes Marciales, enfrentándose a Ten Shin Han. Constará de dos niveles: el primero funcionará como una fase preliminar del combate, donde ambos personajes demostrarán sus habilidades básicas de lucha.

Para avanzar al siguiente nivel, el rival debe alcanzar un nivel bajo de vida. Al cumplirse esta condición (el porcentaje exacto lo definiremos más adelante), el oponente entrará en su segunda fase, lo que nos llevará al siguiente nivel del juego.

**Físicas que utilizaremos en el proyecto:**

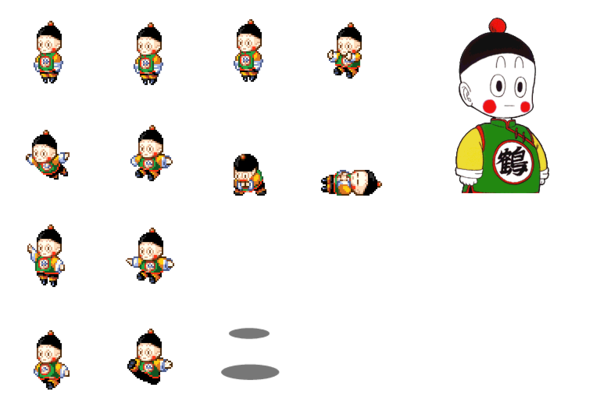
**Movimiento acelerado:** El personaje podrá aumentar su velocidad al presionar dos veces la tecla asignada para esta acción.

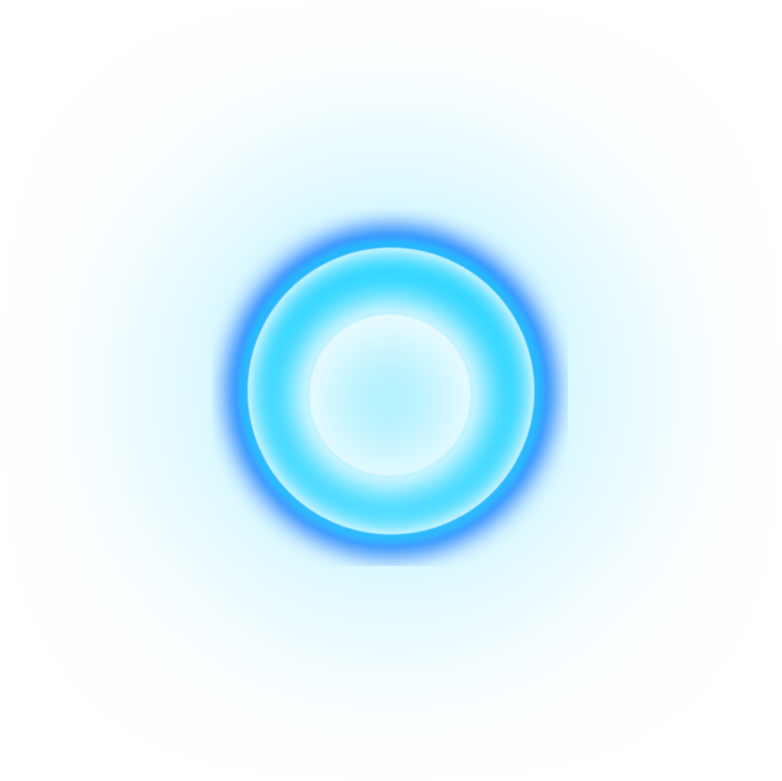
**Movimiento parabólico:** Al lanzar poderes de ki, estos se desplazarán en trayectoria curva hacia el enemigo.

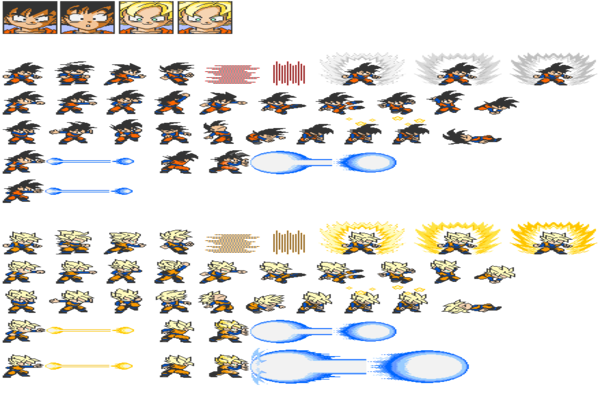
**Colisión:** Cuando un personaje reciba un ataque de ki (tanto básico como especial), sufrirá daño como consecuencia de la colisión.

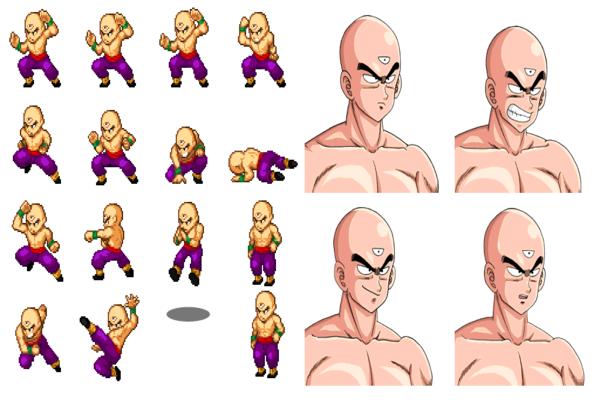
Momento II:

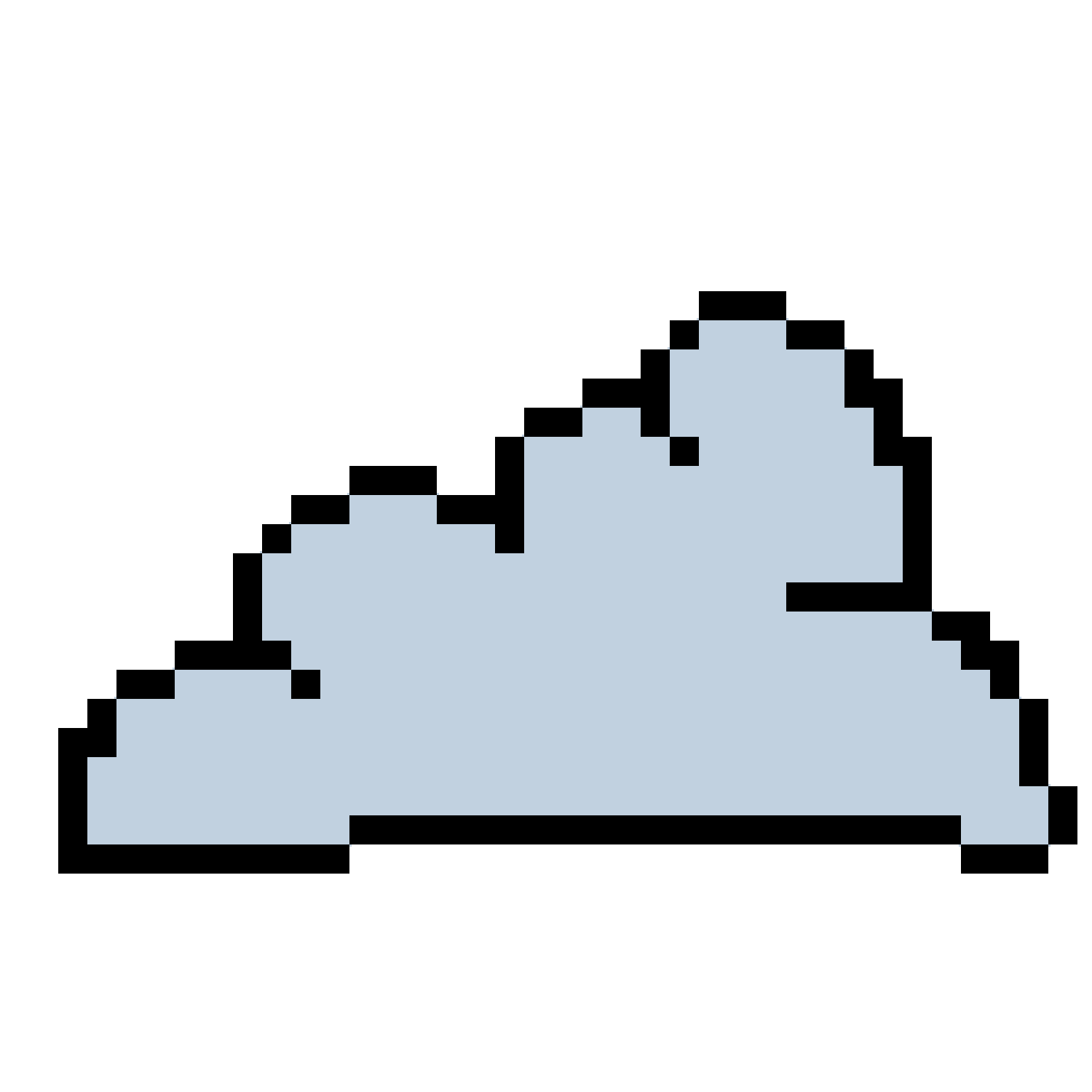
Sprite

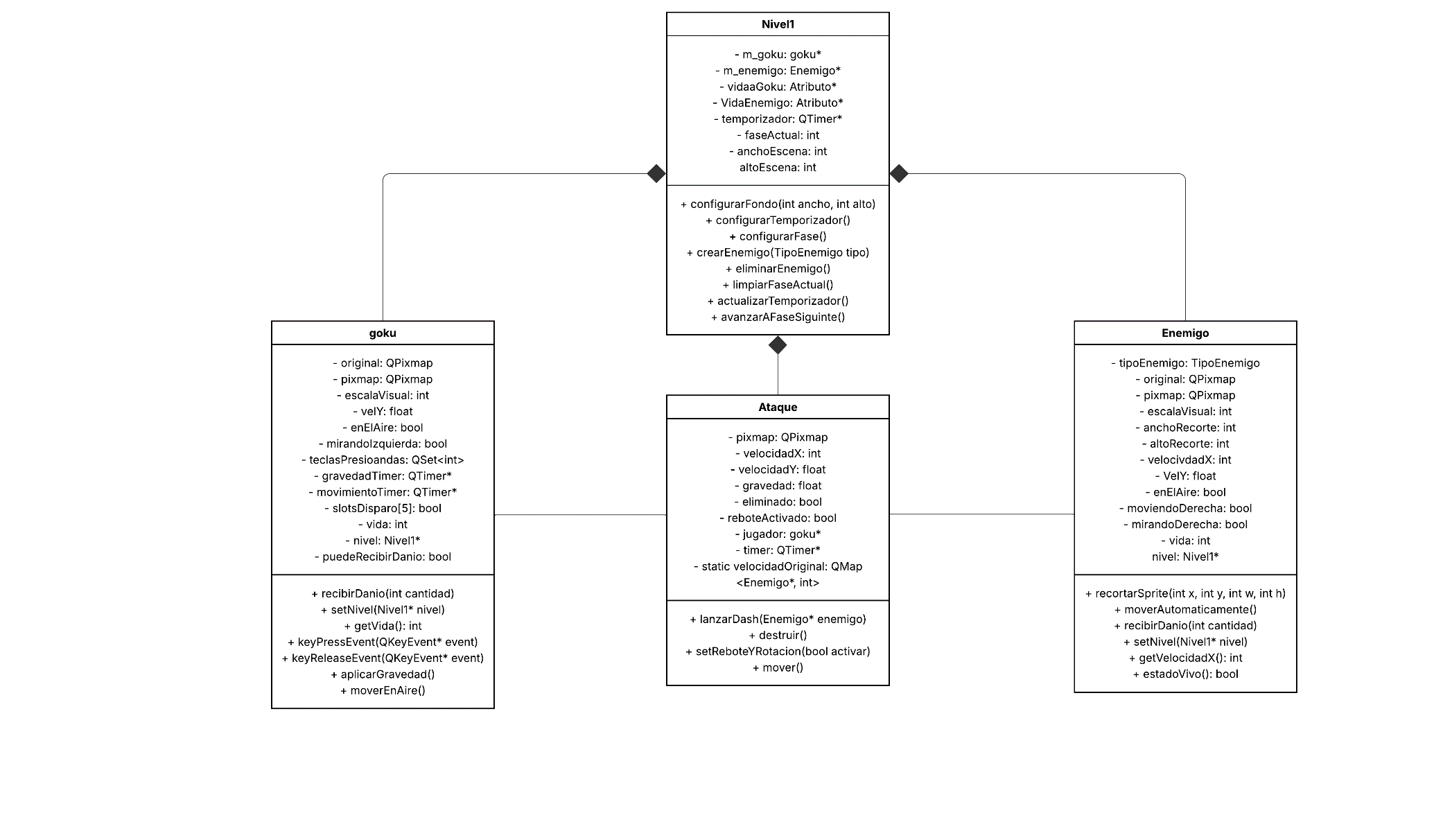












**Tipos de relaciones:  
La composición se representa con un rombo lleno en el extremo de la línea que conecta un objeto con sus partes, indicando una relación fuerte y de dependencia donde la existencia de las partes depende de la existencia del objeto contenedor.**

**Asociación**

**es una relación estructural entre dos o más clases que indica que las instancias de esas clases están conectadas de alguna manera. No implica una dependencia estricta, sino una conexión lógica. Se representa visualmente con una línea que conecta las clases involucradas.**

Nivel1

Atributos: contiene punteros a goku, Enemigo, dos barras de vida (Atributo\*), un QTimer\* y parámetros de escena (ancho, alto, fase Actual).

Métodos: inicializa fondo, temporizador y fase; crea y elimina enemigos; limpia fases; actualiza el tiempo y avanza de fase.

Relaciones:

* (composición) con goku y con Enemigo (Nivel1 “posee” esas instancias).
* Con Ataque (Nivel1 coordina el ciclo de vida de los ataques).

goku

Atributos: Sprite, estado de teclas, temporizadores de gravedad y movimiento, vida, puntero a su Nivel1.

Métodos: responde a teclas (mover, saltar, disparar), recibe daño, establece la referencia a la escena.

Relación:

Con Nivel1 (goku sabe en qué nivel está).

Utiliza Ataque (genera ki).

Enemigo

Atributos: tipo, Sprite, dimensiones de recorte, velocidad, dirección, vida y puntero a Nivel1.

Métodos: recorta su Sprite, se mueve automáticamente, recibe daño, establece su nivel y comprueba si sigue vivo.

Relación:

Con Nivel1 (igual que goku).

Puede lanzar Ataque (dash u otros) sobre goku.

Ataque

Atributos: pixmap, velocidades X/Y, gravedad, estado de rebote, punteros a goku (para el ki) y timers, más un mapa estático de velocidades originales de cada enemigo.

Métodos: lanzarDash (), destruir (), setReboteYRotacion (), y mover ().

Relación:

Asocia tanto con goku (sus ki) como con Enemigo (sus dash).

Cambios del Momento I

Agregamos enfriamientos a las habilidades de goku para añadir una mayor dificultad al juego, como implementamos un cronometro que en cada nivel se va disminuyendo y aumentando la dificultad de cada jefe. Sus respectivos atributos de vida en cada clase enemigo y goku. También en el nivel 2 implementamos ataques del ambiente de ráfagas de ki super rápidas en una parte del mapa para que el jugador tuviera más cuidado al estar, pero también implementaríamos nubes para que el jugador pueda evitar los dash y las ráfagas de ki rebotarían en esta el personaje le ayudaría bastante esto para poder lograr vencer el nivel.

También poder agregar aleatoriamente que salgan semillas del ermitaño en cada nivel para recuperar 1 de vida esto ayuda a nivelar la dificultad del juego aun pensamos en implementar más cosas al código

Cambios del Momento II:

Ya en este momento tuvimos que solo implementar los enemigos, los ataques de ki, agregar la liberación de memoria al codigo fuente para evitar los problemas de crash que se mantuvieron quitamos el Sprite de nube y semillas al no estar bien optimizados agregamos la física de rotacional, parabólica y de rebote que son las que si son aceptadas no las de movimiento rectilíneo que es caída entre otras.