

Información técnica

Animaciones

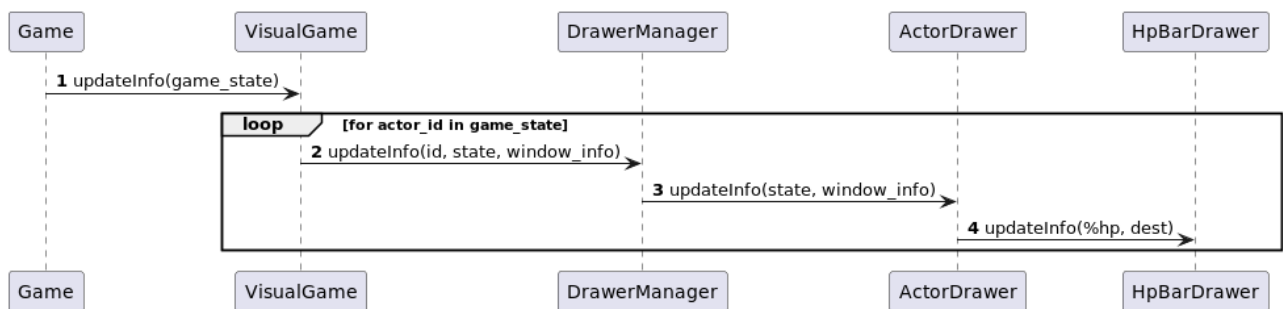
Las animaciones se cargan primero para luego dibujarse.

Las clases encargadas de cargar y preparar las texturas son las Animation.

Las clases encargadas de guardar un estado y luego dibujar son las Drawer.

Este estado se puede actualizar y tiene efectos al momento de dibujar.

Tener en cuenta que actualizar el estado y dibujar son metodos separados.

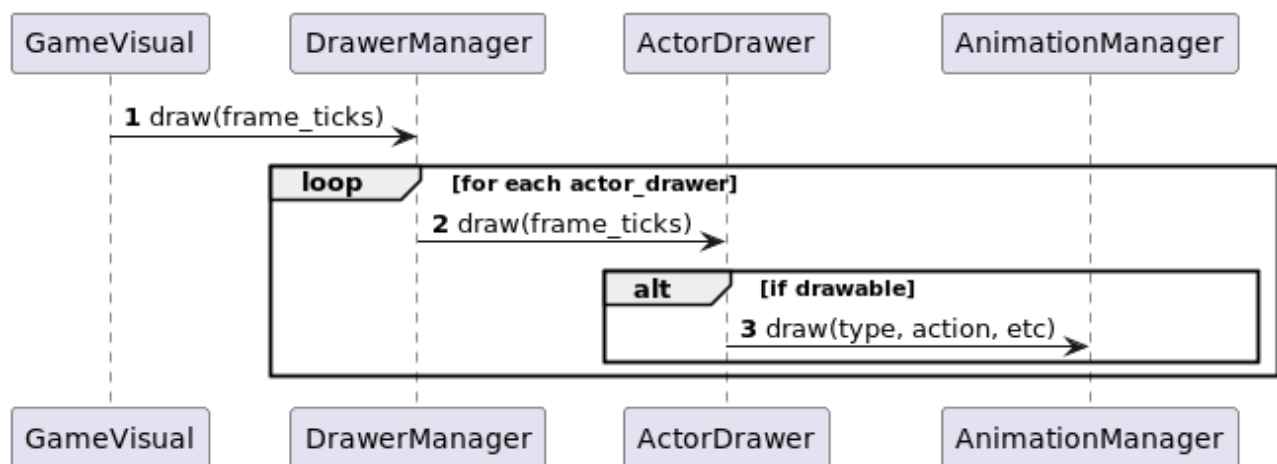


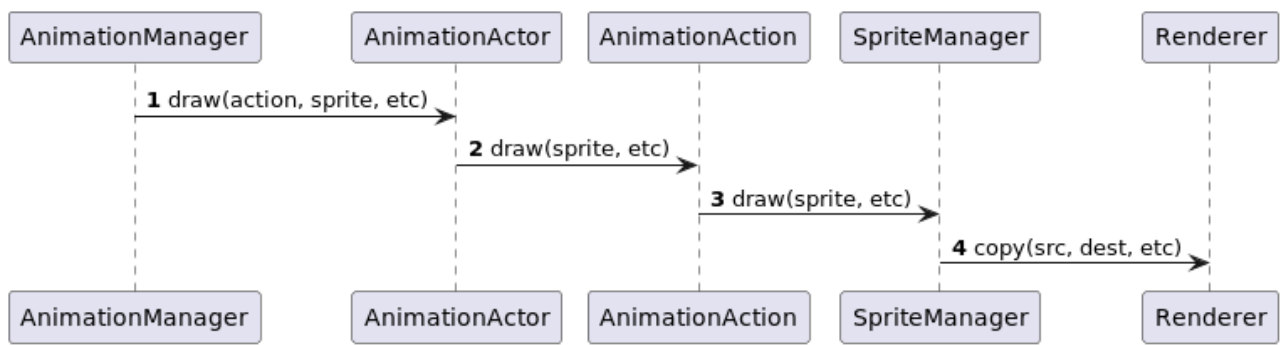
Actualizar la informacion determina la posicion del actor y de su barra de vida, el tipo de accion que debe dibujar, etc.

Al iniciar el programa primero se inicia el manejador de dibujadores quien instancia las animaciones.

Cuando se agregan nuevos actores se crean nuevos dibujadores que hacen uso de las animaciones ya creadas.

Las clases Drawer (dibujadores) delegan el dibujado a las Animation para respetar el encapsulamiento.





Un detalle es que el dibujador guarda el indice del sprite a dibujar pero el sprite manager es quien lo actualiza dependiendo de la velocidad de cada sprite, los milisegundos (frameticks) que pasaron y por cual frame va. Por ende se pasa la direccion de memoria de este indice y sprite manager recibe un puntero hacia ese indice.

```

void SpriteManager::draw(SDL2pp::Texture &texture, std::uint8_t *sprite_index, std::uint8_t direction,
                        const SDL2pp::Point &sprite_destination, std::uint32_t frame_ticks) {
    if (frame_ticks > ms_to_change) {
        *sprite_index = loop_type.nextSprite(*sprite_index, max_index: sprites.size() - 1);
    }
}

```

```

void ActorDrawer::draw(std::uint32_t frame_ticks) {
    std::uint8_t last_sprite = sprite_index;

    if (!drawable)
        return;

    animation_manager.draw(actor_index: type, animation_index: animation, &sprite_index, direction,
                          sprite_destination, frame_ticks: frame_ticks - previous_frame_ticks);
}

```

El tipo de loop decide si la animacion se queda en el ultimo frame (por ejemplo, cuando un actor muere) o si vuelve al primero.