

Documento avance semana 6



El enlace a nuestro repositorio es el siguiente:

<https://github.com/Laurarestrepo03/Ocarina-Galaxian>

Avance

Issues terminados

Esta semana nos enfocamos más en correcciones, por lo que los issues terminados fueron los siguientes:

<input type="checkbox"/>	<input type="radio"/> 0 Open <input checked="" type="radio"/> 7 Closed	Author	Label	Projects	Milestones	Assignee	Sort
<input type="checkbox"/>	<input checked="" type="radio"/> [Fix] Ajustar velocidad de estrellas en pausa y velocidad de jugador fix						
	#48 by Laurarestrepo03 was closed now						
<input type="checkbox"/>	<input checked="" type="radio"/> [Fix] Ajustar sistema de movimiento de enemigo según recomendaciones fix						
	#43 by Laurarestrepo03 was closed 4 days ago						
<input type="checkbox"/>	<input checked="" type="radio"/> [Fix] Crear componente y sistema de parpadeo fix						
	#42 by Laurarestrepo03 was closed yesterday						
<input type="checkbox"/>	<input checked="" type="radio"/> [Fix] Ajustar funcionamiento del fondo según recomendaciones fix						
	#41 by Laurarestrepo03 was closed yesterday						
<input type="checkbox"/>	<input checked="" type="radio"/> [Fix] Sistema unificado para bala del jugador fix						
	#39 by Laurarestrepo03 was closed 4 days ago						
<input type="checkbox"/>	<input checked="" type="radio"/> [Fix] Crear estrellas como cuadrados fix						
	#35 by Laurarestrepo03 was closed now						
<input type="checkbox"/>	<input checked="" type="radio"/> [Obligatorio] Despliegue del texto de puntaje en la parte superior con su texto 1-UP enhancement						
	#9 by lacortez was closed 3 days ago						

Notas: Todavía no hemos implementado que se elimine la entidad del jugador cuando colisiona con una bala, ni el delay en el disparo de los enemigos.

Componentes

Al lado de cada componente se encuentra una **A**, **D** o **U**. Estas letras hacen referencia, respectivamente, a si se añadió, eliminó o actualizó el componente. Se omiten los componentes que permanecieron igual:

- **U** CTagBullet: Este componente ahora tiene un atributo que indica si la bala es del jugador o enemiga. Esto es porque se volvió obsoleto el uso de etiquetas separadas para ambos.
- **D** CTagEnemyBullet: Se eliminó por las razones mencionadas anteriormente.
- **U** CTagPause: Se eliminó el atributo de temporizador que se usaba para el parpadeo, pues ahora este es un componente aparte.
- **D** CTagPlayerBullet: Se eliminó por las razones mencionadas anteriormente. Además, el atributo que tenía, que indicaba si ya se había disparado, ahora hace parte de un componente nuevo.
- **A** Cblink: Para gestionar la información relacionada al parpadeo de una entidad.
- **A** CPlayerBulletState: Se agregó para gestionar el estado de la bala del jugador, es decir, si no se ha disparado o si está en el aire.

- **D** CStarField: Se eliminó según las recomendaciones dadas.

Sistemas

Se usan las mismas convenciones **A**, **D** o **U** de los componentes. Se omiten los sistemas que permanecieron igual:

- **A** system_blink: Para hacer que las entidades con el componente CBlink parpadeen. Funciona modificando la superficie de la entidad.
- **U** system_bullet_limit: Se editó para adaptarse a los cambios de la bala del jugador, porque ahora no se elimina esta entidad cuando llega al límite, simplemente se cambia su posición. Este cambio hace que sea necesario diferenciar si la bala es enemiga o del jugador: si es del jugador se cambia su estado a *NOT_FIRED*, y si es enemiga sí se elimina.
- **D** system_collision_bullet_enemy: Se eliminó porque ahora se maneja en el sistema de estado de la bala del jugador.
- **D** system_draw_stars: Se eliminó según las recomendaciones dadas.
- **U** system_enemy_movement: Se eliminó la línea que estaba causando que se actualizara la posición también en este sistema.
- **D** system_player_bullet_rest_pos: Se eliminó porque ahora se maneja en el sistema de estado de la bala del jugador.
- **D** system_player_bullet_spawn: Se eliminó porque ahora se maneja en el sistema de estado de la bala del jugador. Es más, ahora no hay un spawn como tal, porque la bala no se vuelve a crear.
- **A** system_player_bullet_state: Para manejar los estados de la bala del jugador. Cuando se encuentra en reposo (*NOT_FIRED*) su posición debería ser justo encima del jugador. Si se encuentra en el aire (*FIRED*), se verifica si colisiona con un enemigo, y si sí, se destruye el enemigo, se crea la explosión, y se envía la información del tipo de enemigo destruido para actualizar el puntaje.
- **U** system_rendering: Como ya no se maneja el parpadeo de la pausa desde acá, se restaura a su estado original.
- **U** system_star_field: Gracias al nuevo sistema de parpadeo, ahora este sistema solo se encarga de que 1) no se pare el movimiento de las estrellas cuando se está en pausa y 2) asignar una nueva posición aleatoria a las estrellas cuando llegan al límite de la pantalla.
- **A** system_update_high_score: Para actualizar el valor del high score si el puntaje actual es mayor. También actualiza este valor en el archivo de configuración.
- **A** system_update_score: Para actualizar el texto del score cada vez que se elimina un enemigo.¹

Funciones prefab

Se usan las mismas convenciones **A**, **D** o **U** de los componentes. Se omiten las funciones que permanecieron igual:

- **U** create_bullet: Se actualizó para que ahora la etiqueta de bala también reciba el tipo (jugador o enemiga). No solo eso, sino que si la bala es del jugador, se agrega el componente CPlayerBulletState.

¹ Como se puede notar, los sistemas de actualizar los scores están creando y borrando entidades, pero lo mejor es simplemente hacer un render. Ya tenemos planeado hacer un issue con este cambio. Igual nos gustaría tener confirmación para saber si este cambio es correcto.

- **U** create_star: Se actualizó para que ahora tenga el componente de parpadeo y se cree con la función de crear cuadrados.
- **U** create_level: Se eliminó una variable que no se usa.
- **D** create_pause_text: Se eliminó, pues ahora se tiene una función genérica para crear texto.
- **A** create_text: Crea entidades de texto.

Archivos de configuración

Se usan las mismas convenciones **A**, **D** o **U** de los componentes. Se omiten los archivos que permanecieron igual:

- **U** enemies: Se actualiza para agregar la cantidad de puntos que da dispararle (en su movimiento normal).
- **U** interface: Se actualiza para agregar los textos de score y 1UP. También almacena el valor del high score que se modifica con el sistema mencionado anteriormente.
- **U** player_bullet: Se aumentó un poco la velocidad de la bala del jugador.
- **U** player: Se redujo un poco la velocidad del jugador.

Otros cambios

Vale la pena mencionar que debido a todos los cambios listados, el archivo game_engine.py también cambió, por lo que recomendamos echarle un vistazo.

Dudas

- Todavía no nos queda muy claro la necesidad de crear un servicio de configuración, dado que con la función de _load_config_files se cargarían los archivos solo una vez, ¿no? También, digamos, si por alguna razón cambia el nombre del archivo, solo habría que editarlo en el load, pero si tenemos un servicio, ¿no habría que cambiar el nombre del archivo en todas las funciones que lo usen? Es decir, ¿no se vuelve un poco impráctico si hay varias funciones usando la misma configuración?
- Estamos trabajando en el feature del texto de READY. No estamos totalmente seguros si vamos por el camino correcto, por lo que pedimos el favor de hacer checkout a la rama feature/8-texto_ready_sonido.
- ¿Qué tan necesario sería implementar el manejo de escenas para el GAME OVER o para avanzar de nivel?
- En general, ¿hubo algo de los cambios y adiciones que hicimos que esté mal o que recomiendan cambiar?

¡Gracias!