Comet strikes back

Memoria Práctica 1 Multimedia

04-03-2020

Sergio Lira Díaz – GIS/GII – 202

Lucía Colás Vico – GIS -1001

Tabla de contenido

[1. Dificultades encontradas y su solución 2](#_Toc34224503)

[1.1. Bugs en el reinicio del nivel 2](#_Toc34224504)

[1.2. Funcionalidad de eliminar el héroe cuando va muy despacio 2](#_Toc34224505)

[1.3. Funcionalidad de propulsión de meteor2 2](#_Toc34224506)

[2. Conclusiones 3](#_Toc34224507)

[2.1. Sergio 3](#_Toc34224508)

[3. Bibliografía 3](#_Toc34224509)

# Dificultades encontradas y su solución

## Bugs en el reinicio del nivel

Se detectaron dos bugs en al reiniciar un nivel no finalizado: el juego se quedaba atascado cargando los assets, y no se inicializaba el héroe en el tirachinas. En el caso de los assets bastó con modificar la condición de carga total para inicializarlo. Por otro lado, el caso del tirachinas ha sido más complejo, pues se ha tenido que depurar el código en profundidad para ver que al reiniciar el nivel a medias, el atributo “game.currentHero” todavía cargaba el héroe de la partida anterior, impidiendo cargar los de la nueva partida. Tras hallar la raíz del problema ha bastado con vaciar currentHero al pulsar el botón de reiniciar.

## Funcionalidad de eliminar el héroe cuando va muy despacio

La mayor dificultad en esta funcionalidad ha sido determinar cómo establecer un contador de tiempo, cómo anexarlo a un evento, etc. Tras buscar documentación se ha implementado un método que, al tener el héroe una velocidad muy baja, dispara un contador mediante el método de JS setTimeOut(), marcando el héroe como destruible, para después ser destruido.

## Funcionalidad de propulsión de meteor2

Esta probablemente ha sido la funcionalidad más compleja, ya que ha exigido una fase de documentación considerablemente más larga. Se deseaba implementar una funcionalidad en la que al pulsar mientras el asteroide estaba volando, obtuviera un impulso extra para chocar con más fuerza en la fortaleza de los villanos. Primero se consideró eliminar la gravedad de esa entidad, sin embargo se descartó ya que, además de no lograr el efecto deseado, también era necesario eliminar la caída del impulso del tirachinas. La segunda opción fue añadirle una propulsión mediante el método de Box2D “applyForce”, propulsándolo en la dirección que tuviese el héroe en el momento de la pulsación. No obstante, ese método no detenía al héroe al chocarse con algo, y empujaba infinitamente en la dirección establecida. Finalmente, se modificó el método “applyFoce” por “applyImpule”, logrando la funcionalidad deseada.

# Conclusiones

## Sergio

Esta práctica me ha enseñado sobre todo a observar y aprender cómo funcionan las físicas de un juego modelado (entidades, colisiones, impulsos, fuerzas, etc) y a interactuar con ellas. No obstante, la librería Box2D me ha parecido un tanto limitada en cuanto a funcionalidades se refiere. Por ejemplo, no proporcionar un método para desactivar la gravedad de una entidad en concreto, y tener que diseñar una fuerza aparte que contrarreste la gravedad estipulada. En resumen, me ha llamado la atención de esta práctica, pero me gustaría investigar otras opciones para desarrollarla.

# Bibliografía

* *Forces and impulses*. Iforce2d: <https://www.iforce2d.net/b2dtut/forces>
* *Physics with Box2D*. ImpactJS: <https://impactjs.com/documentation/physics-with-box2d>
* *Box2d and constant movement.* StackExchange: <https://gamedev.stackexchange.com/questions/24453/box2d-and-constant-movement>
* *JavaScript Timing Events.* W3schools. <https://www.w3schools.com/js/js_timing.asp>