Acerca del

Videojuego

usando el

Motor Phaser

y el sistema de control de versiones

Git

y almacenado en

GitHub*

trabajo práctico para

Programación Orientada A Objetos

Profesor: Mariano D'agostino

F Emmanuel Messulam Franco I Vallejos Vigier

Instituto Politécnico Superior de Rosario "Gral. San Martin"

*https://github.com/EmmanuelMess/School-Phas3r-Game-Project

Introducción

Se presenta el trabajo practico, un juego donde el jugador tiene que luchar contra zombies para pasar el nivel. El nivel de dificultad es progresivo, es decir no se empieza con una oleada de 15. El nivel de dificultad crece exponencialmente, véase el array `EnemigosACrear` en el archivo "Principal.js".

A pesar de que el juego no posee tutorial, creemos intuitiva la forma de interaccion, (las teclas WASD para moverse, que son ampliamente usadas, y el espacio para disparar son una de las primeras cosas que, creemos, el jugador va a intentar).

Desafios

Desafíos no se nos presentaron en la forma en la que normalmente aparecen en proyectos con cientos de miles, o millones, de lineas. Si bien hubo que lidiar con el desconocimiento general de algunas herramientas proveídas por la plataforma GitHub, como la "Pull request", que no existe en git.

No se presento mayor dilema a la hora de separar el proyecto en archivos, ya que no iba a ser mantenido por un largo periodo de tiempo, se decidió poner todo en un solo archivo "Principal.js".

Tampoco se presento mayor dilema para separar todo en clases, ya que las ideas de "objeto" son muy fáciles de implementar en un videojuego, cosas como una 'bala' o un 'zombi' son claramente clases; y, claramente, el juego en sí es un objeto del cual solo existe una instancia, "singleton". Aunque esta ultima no esta expresada en código como tal, el juego es simplemente un diccionario de funciones, por la forma en que Phaser trata al juego y como ECMAScript trata a las clases y objetos.

Ejemplos

Instancia de clase

Método publico

Método privado

Clase

Atributo de clase

Constructor

Mejoras posibles

Para una guía más actualizada y más completa véase la pagina de issues (filtrando por "enhancements") en GitHub[1].

Un sistema de niveles

Más niveles, un menú con más niveles. Un sistema de progreso donde el jugador pueda avanzar y enfrentarse a enemigos más fuertes.

Un mecanismo de experiencia

Un sistema de experiencia con contenido desbloqueable para otorgar más fuerza en los ataques. Mejoras graduales al jugador ayudarian a contrarrestar los enemigos más fuertes si se implementaran más niveles.

Actualizar a la versión 3 del motor Phaser

Como dice el FAQ (preguntas frecuentes, por sus siglas en ingles) de Phaser 3 (traducción)[2]:

> ...como v3 otorga un montón de nuevas formas de lograr objetivos comunes. Te darán la posibilidad de que tu código sea más claro y más estructurado. Aprovechalo.

Pero aún hay pocos ejemplos y documentación con respecto a la nueva versión, por lo que se prefirió usar la v2. En un futuro debería moverse el proyecto para aprovechar la nueva versión.

[1] `https://github.com/EmmanuelMess/School-Phas3r-Game-Project/issues?q=is%3Aissue+is%3Aopen+sort%3Aupdated-desc+label%3Aenhancement`

[2] Véase la pregunta tres en `https://phaser.io/phaser3/faq`