

Introducción a la Programación de Videojuegos y Gráficos

GALACTIC STRIKE (provisional)

Realizado por:

Eduardo Radío Gallego, eduardo.radio@estudiante.uam.es

Borja Mauricio Fourquet Maldonado, borja.fourquet@estudiante.uam.es

Fecha:

1/11/2015

Introducción a la Programación de Videojuegos y Gráficos

1. Descripción general del proyecto

En esta sección se deben especificar los objetivos generales del proyecto, así como una descripción del equipo disponible y el entorno de desarrollo que se va a emplear. Se describen las principales técnicas a emplear. Se describirá de forma general la funcionalidad del proyecto, posibles requisitos hardware, así como cualquier otra descripción o recurso adicional que sea de interés a un nivel más general.

Este punto se podría subdividirse (entre otras) en las siguientes secciones:

1. Introducción

El objetivo principal del proyecto es el de desarrollar un videojuego de navegador en línea, novedoso y divertido, aunando características de diversos tipos de juegos.

2. Descripción general del producto (en este caso un videojuego).

Galactic Strike es un juego multijugador de peleas por equipos en el espacio. Los jugadores elegirán o crearán un escenario con un número determinado de jugadores, seleccionarán uno de los distintos personajes disponibles, se unirán a un equipo y competirán contra otros en distintos modos de juego. Durante el combate aparecerán distintos objetos aleatoriamente que los personajes podrán utilizar a su favor para dominar la partida.

3. Funcionalidad (reglas del juego, objetivos, mundo, etc).

El juego será esencialmente **PVP** (*Player Versus Player*).

En el principal modo de juego el objetivo de cada equipo será aniquilar a los miembros del equipo contrario. A medida que se avance con el desarrollo del juego, podrían incluirse distintos formatos de partida (aniquilación por rondas, capturar la bandera, todos contra todos, etc.).

El mundo en el cual se llevarán a cabo estas batallas estará compuesto por diversos astros con distintas características. Los jugadores tendrán que saltar de un planeta a otro para recolectar objetos, alcanzar a sus enemigos, huir, etc. Un escenario estará compuesto por un conjunto de astros estáticos, que normalmente generarán un campo gravitatorio que atraerá a cada objeto o personaje que entre en él, además de otro posible conjunto de astros móviles espontáneos, como meteoritos o agujeros negros, que jugarán un papel de enemigo neutral.

Introducción a la Programación de Videojuegos y Gráficos

4. Dominio, o género, de desarrollo del proyecto.

Se trata de un videojuego plataformas en 2D, con ideas extraídas del clásico Counter Strike [3], pero utilizando personajes más propios de un juego de rol, y todo ello en un escenario más propio de un “Space Ship Game” [4] o juego de naves espaciales.

5. Entorno y herramientas de desarrollo.

Se trabajará sobre el sistema operativo **Windows 7**. Usaremos el IDE **Brackets**, y probaremos el software en el navegador **Google Chrome** (en la última versión estable que exista en el momento subida de cada uno de los entregables). El juego se codificará en **JavaScript** sobre **HTML5**. Se usará el *framework* **Phaser** tanto en el lado del cliente como en el del servidor para gestionar las físicas, los gráficos, etc. Además, para gestionar las conexiones, las llamadas a métodos remotos y demás se utilizará algún *framework* que se encargue de ello, como **Pomelo** [2] o **Eureca** [5], que utilizan **NodeJS**.

6. Descripción del hardware.

El videojuego se desarrollará principalmente para los ordenadores de la EPS, los cuales en su mayoría son de gama media-baja y cuentan con procesadores de doble núcleo.

7. Equipo y lugar de trabajo.

El equipo está compuesto por dos personas, que realizarán labores tanto de diseño como de implementación:

Eduardo Radío Gallego: eduardo.radio@estudiante.uam.es
Borja Mauricio Fourquet Maldonado: borja.fourquet@estudiante.uam.es

8. Técnicas a emplear.

Utilizaremos una metodología ágil basada en el archivo adjunto GS-CR-vX.xlsx, que se irá completando a lo largo del desarrollo.

Los miembros del equipo se irán autoasignando requisitos de este documento según su prioridad, y al final volverán a seleccionar otro requisito que llevar a cabo. Los requisitos “extra” los proponemos para la entrega final en el caso de que hubiésemos completado todos los requisitos anteriores (“prototipo” y “final”).

9. Recursos adicionales (por ejemplo, gráficos, sonidos, etc).

Para la primera versión utilizaremos mayormente sprites y sonidos de terceros, aunque si se da bien para la última entrega, se pretende incorporar gráficos y audios de elaboración propia.

Introducción a la Programación de Videojuegos y Gráficos

2. Gestión del Proyecto

En esta sección se deben mostrar las planificaciones temporales de desarrollo del proyecto en su fase de inicio y de elaboración, así como el diario de ejecución del proyecto, junto con el diario de construcción de la aplicación y cumplimiento de los plazos estimados.

1. **Entregables** (en el caso de la asignatura 3, incluyendo este documento¹) y fechas de los mismos.

❖ *Práctica 1, Domingo 1 de Noviembre de 2015:*

- *Documento de Análisis y Diseño (este mismo documento): GS-DAyD-v1.0.pdf*
- *Diagrama de Clases: GS-DC-v1.0.pdf*
- *Diagrama de Estados: GS-DE-v1.0.pdf*
- *Catálogo de Requisitos: GS-CR-v1.0.xlsx*

❖ *Práctica 2:*

❖ *Práctica 3:*

2. **Contenido** de cada entregable. Tanto en forma de documentación, como de software o prototipos a mostrar en cada uno de ellos.

Matriz de trazabilidad de cambios

Fecha	Identificación del Cambio	Documento de Análisis y Diseño	Diagrama de Clases	Diagrama de Estados	Catálogo de Requisitos
1/11/2015	Entrega práctica 1	GS-DAyD-v1.0.docx	GS-DC-v1.0.html	GS-DE-v1.0.html	GS-CR-v1.0.xlsx
	Entrega práctica 2				
	Entrega práctica 3				

¹ En el caso de este proyecto, la fecha límite del primer entregable será el **1 de Noviembre de 2015**

Introducción a la Programación de Videojuegos y Gráficos

3. Análisis/Diseño

En esta sección se describirán los modelos de análisis/diseño (diagrama de clases, si se usa programación orientada a objetos, patrones de diseño) como el modelo de datos (e.g. modelo entidad - relación), desde los cuales se puede consultar la especificación de los métodos de clase más relevantes o las especificaciones de atributos.

También es deseable proporcionar un diseño inicial (aunque no sea definitivo) de una propuesta de diseño de clases (interfaces, clases abstractas, patrones de diseño a utilizar,...) que pudiesen emplearse para el posterior desarrollo del proyecto.

4. Implementación

En esta sección se muestran, o se describen, los prototipos de interfaces de usuario de la aplicación, así como un diagrama de la funcionalidad añadida en cada fase de producción.

5. Pruebas

Por último, se describirán las pruebas a realizar en cada iteración de la aplicación.

6. Referencias

[1]: https://es.wikipedia.org/wiki/Videojuego_de_plataformas

[2]: <http://pomelo.netease.com/index.html>

[3]: <https://en.wikipedia.org/wiki/Counter-Strike>

[4]: <http://es.y8.com/tags/spaceship> - Ejemplos de Spaceship Games

[5]: <http://eureca.io/>