

Plan de trabajo

Descripción general del trabajo, con una lista de objetivos.

El trabajo consiste en el estudio y desarrollo de una aplicación que permite la reproducción de un tipo específico de videojuegos usando un intérprete genérico para generar los eventos del mismo. Para ello se seguirán los pasos que se describieron en la propuesta:

1. Análisis de anteriores implementaciones del motor.
2. Definición de la arquitectura del desarrollo.
3. Diseño de sistema de codificación de los ficheros de configuración de un videojuego.
4. Desarrollo de motor genérico que interprete ciertos ficheros y los transforme en un modelo reproducible por un videojuego.
5. Desarrollo de la lógica genérica del videojuego y su interfaz gráfica reproducible según los ficheros de configuración.
6. Crear videojuego de ejemplo.

Durante todas estas secciones se irán documentando los nuevos conceptos que aparezcan, por lo que se añade un día extra a cada tarea, tres en el caso del punto cinco.

En este proyecto se busca focalizar el trabajo en el desarrollo del motor genérico de interpretación de ficheros. Esto se quiere llevar a cabo siguiendo una metodología de trabajo ágil, específicamente dentro del marco SCRUM, por lo que antes de redactar este documento se ha hecho un “*sprint backlog*” general de todo el desarrollo.

Debido a que todavía no se ha valorado exactamente cuánto se va tardar en cumplir los requisitos, es posible que alguna de estas historias de usuario no se llegue a completar o que haya la posibilidad de implementar funcionalidades extra.

Asimismo, estas historias de usuario representan requisitos amplios, por lo que es probable que se subdividan en varias historias de usuario más específicas; y no están ordenadas, por lo que puede que no se desarrollen en el mismo orden que se presenta.

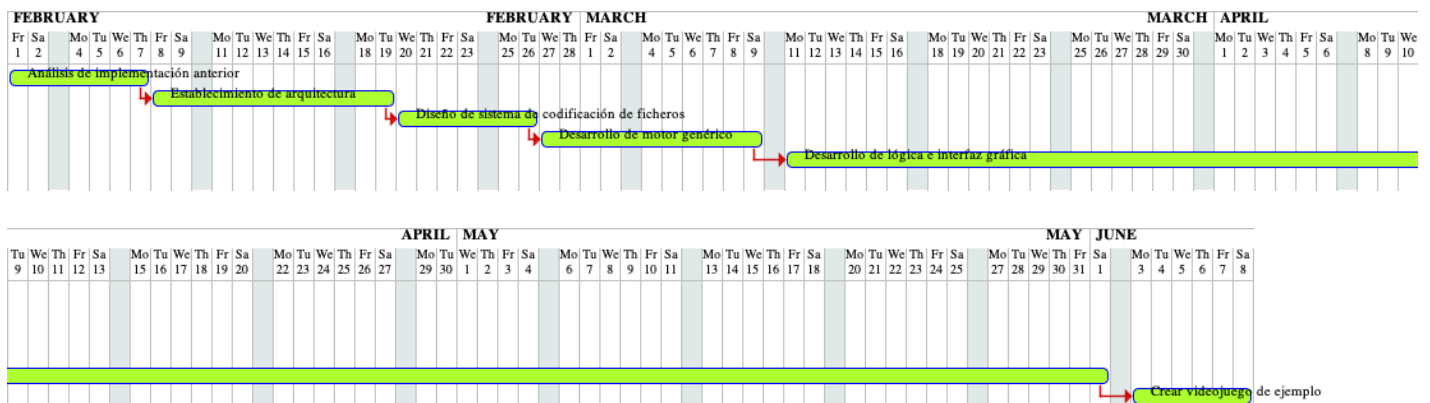
Se pretende fijar ciertas tareas para cada “*sprint*” para poder llevar una documentación del proceso de desarrollo.

Historias de usuario:

- Diseño de los ficheros de configuración y descodificación en el programa: implementar medios para soportar la lectura e interpretación de los ficheros.
- Menú de inicio: menú principal desde el que aparece el juego.

- Pantalla de juego: pantalla principal donde se centra el desarrollo del juego.
- Menú de movimiento: menú que permite al personaje moverse entre salas del juego.
- Menú principal: menú disponible durante el transcurso del juego, con múltiples opciones.
- Menú de inventario: menú que permite interactuar con el inventario del personaje principal.
- Diálogos intermedios: sistema de diálogos entre los personajes.
- Acciones genéricas de la pantalla de juego: eventos que ocurren al pulsar los botones de acción disponibles.
- Sistema de batalla: evento específico que representa una batalla entre personajes.

Diagrama de Gantt



Propuesta de TFG

	10MI, 10II
ID de la oferta	3473
Título del trabajo	Historias del laberinto
Tipo de trabajo	
Resumen general del trabajo	Desarrollo de un programa que soporte reproducción de videojuegos realizados en archivos de configuración en una plataforma accesible.
Lista de objetivos concretos	<ul style="list-style-type: none"> - Introducción de desarrollo llevado a cabo hasta el momento. - Definición del lenguaje óptimo a utilizar en el proyecto. - Desarrollo de un motor genérico de creación de videojuegos con una arquitectura limpia que sea fácilmente mantenible. - Análisis de la posibilidad de desarrollar un sistema con interfaz gráfica. - Añadido de una configuración estándar para ejemplificar las posibilidades del juego.

Desglose de la dedicación total del trabajo en horas (324 horas en los Grados)	<ul style="list-style-type: none"> - Análisis de elección de lenguaje óptimo para programar. (10h) - Establecimiento de arquitectura a seguir. (15h) - Migración y refactorización a nueva arquitectura. (100h) - Pruebas con la migración. (20h) - Implementación de nuevas funcionalidades. (100h) - Proceso de pruebas. (40h) - Documentación (40h)
Conocimientos previos recomendados para hacer el trabajo	<ul style="list-style-type: none"> - Necesario: Git, Java - Muy recomendable: Swift
Fichero con los detalles del trabajo	
Tutor 1	ÁNGEL HERRANZ NIEVA - aherranz@fi.upm.es