

Videojuego para que los niños aprendan a programar.

Grado en Ingeniería Informática



Trabajo Fin de Grado

Autor:

Juan Esquerdo Roig

Tutor/es:

Rosana Satorre Cuerda

Carlos Villagrá Arnedo

Enero 2014

**Videojuego para que los niños aprendan a programar.**

UNIVERSIDAD DE ALICANTE

CURSO 2013-2014

Tutores

**Rosana Satorre Cuerda**

**Carlos Villagrá Arnedo**

Alumno

**Juan Esquerdo Roig**

**Índice general**

**Capítulo 1**

**Introducción**

* 1. **Aprender a programar**

Los lenguajes de programación han formado parte de los ordenadores desde sus inicios, concretamente Fortran fue el primer lenguaje de programación ampliamente conocido y exitoso (desarrollado entre 1954 y 1957). Fortran es un lenguaje de programación de alto nivel de propósito general, procedimental e imperativo, que está adaptado al cálculo numérico y a la computación científica.

Más adelante nacerían distintos lenguajes de programación con distintas características. Pero todos ellos tienen la finalidad de crear programas o tareas que exhiban un comportamiento deseado; para por ejemplo, resolver problemas numéricos; de aquí la importancia de aprender a utilizarlos.

Aprender a programar requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar. Es por ello que aprender a programar puede ser una ardua tarea sino se dispone de los recursos correctos.

Actualmente se dispone de distintos recursos para el aprendizaje de la programación, se sigue disponiendo de los recursos tradicionales (libros, clases); pero el aprendizaje de los lenguajes de programación (y otras ramas) está tendiendo hacia recursos más interactivos; ya que si no tenemos metas concretas, o algún aliciente que nos anime a seguir, es posible que abandonemos el aprendizaje.

Podemos recurrir a métodos divertidos y estimulantes para aprender como por ejemplo aprender jugando. Y es que podemos encontrar juegos que nos permiten mejorar nuestras habilidades como programadores.

Una de las principales razones de la investigación y desarrollo de estos recursos interactivos (como por ejemplo los videojuegos) es debido a que desde una edad temprana puede resultar más fácil aprender el concepto abstracto de la programación; resultando para un niño más divertido el aprendizaje de los lenguajes de programación mediante un videojuego.

En la web podemos encontrar videos explicando distintos lenguajes de programación, páginas interactivas donde podremos comprobar las sentencias que estamos aprendiendo, cursos online y un largo etc. Pero estos recursos pueden llegar a resultar un poco tediosos y aburridos para un niño.

* 1. **Objetivos del proyecto**

Se desarrollará un videojuego para PC para que los niños aprendan a programar, y se centrará principalmente en el aprendizaje de diferentes estructuras de programación en diferentes partes del proyecto, aunque no es competencia de este proyecto el que el juego esté completo en su totalidad, si no que sea jugable y pueda mostrar los elementos que lo componen como la interfaz usuario y principalmente el aprendizaje de programación del juego, para lo que se usarán ciertos elementos como sonidos, música o fondos de pantallas que no son propiedad del autor de este trabajo de final de carrera sino que son de licencia “Creative Commons No Comercial” y que, en caso de usarse una parte o totalidad del presente proyecto en futuro proyectos comerciales, deberán ser removidos para evitar problemas legales.

Para lograr el objetivo, diseñará e implementará un sistema de interfaz GUI que permitirá al usuario interactuar con el videojuego y viceversa mediante sus elementos y navegar entre diferentes niveles y que será reutilizable en futuros proyectos en los que sea necesaria una interfaz de usuario en entornos gráficos.

Podremos elegir entre distintos tutoriales donde aprenderemos las sentencias básicas de programación y además un nivel final donde se aprenderán más características de las sentencias básicas; y la combinación de varias sentencias.

Habrá un personaje principal que controlará el usuario mediante las flechas y que al chocarse con diferentes obstáculos, podrá eliminar los obstáculos escribiendo la sentencia de programación correcta para ese evento; pasando a la selección de un nuevo nivel o avanzando en el nivel final, creando una sensación de avance progresivo durante el transcurso de la partida; además de adquirir el conocimiento de la sentencia que necesitara para posteriores obstáculos.

Para la comprobación de que la sentencia escrita por el usuario es correcta, se ha utilizado cadenas de texto; independientemente de los espacios, tabulaciones o saltos de línea que el usuario escriba; ya que cada usuario tiene una manera personal de escribir el código. Además mediante expresiones regulares y atributos de los String se consigue avisar al usuario si ha cometido un error y especificar en algunos casos que error (Olvidado un paréntesis, olvidado llaves).

* 1. **Metodología**

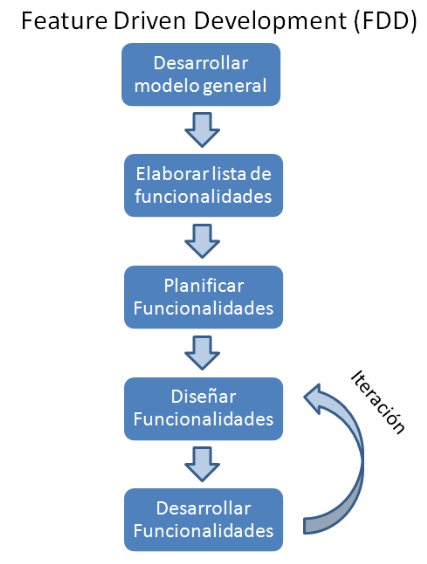
El desarrollo de un videojuego engloba ciertas limitaciones en la planificación como definición de requisitos débiles o poco específicos, largos periodos de construcción del proyecto que dificultan la planificación temporal, la necesidad de mucho aprendizaje y la incertidumbre de si ciertas partes van a ser viables o no al inicio. Por todo esto se ha decidido usar como metodología de trabajo una metodología ágil ya que se ajusta con la mayoría de restricciones comentadas.

Dentro de las metodologías ágiles hay varias opciones disponibles como Scrum, xP, ASD, FDD y muchas otras, pero se ha seleccionado FDD (Feature-Driven Development), debido a que el juego dispone de muchos niveles distintos y se pueden ir añadiendo al sistema como features en cada iteración.

Hay que tener en cuenta que aunque se haya usado una metodología ágil, estas suelen estar preparadas para trabajo en equipo, y dado que este trabajo se ha desarrollado de manera individual no se han seguido todas las características de la metodología al pie de la letra.

La metodología es iterativa e incremental y se compone de 5 fases, 3 de ellas iniciales en las que se crea una visión general del proyecto y se obtiene una lista de funcionalidades ordenadas por prioridad y las dos últimas fases en las que se itera por cada una de las funcionalidades de la lista y en las que se diseña y se implementa esa funcionalidad.

Las reuniones con el cliente (en este caso se entiende como ‘cliente’ a los tutores del TFG) se han realizado en periodos de 1 semana en los que se mostraban los resultados finales de la funcionalidad programada para la iteración y se recogía el feedback para modificar si fuera necesario y posteriormente se iniciaba la siguiente iteración hasta que el producto estuviera terminado.



*Fases de la metodología FDD*

La aplicación de la metodología puede observarse con más detalle en el capítulo 10, donde se analizan las horas dedicadas al proyecto, así como su distribución por módulos que han servido como base para la organización a base de features de la metodología.

**Capítulo 2**

**Estado del arte**

* 1. **Juegos para aprender a programar**

Hablar de los distintos juegos para aprender a programar.

* 1. **Juegos educativos**

Hablar de distintos juegos para aprender a programar

* 1. **Partes**

**Capítulo 3**

**Requisitos**

* 1. **Requisitos funcionales**

Distintos requisitos funcionales.

* 1. **Requisitos no funcionales**

Requisitos no funcionales (Que funcione en Windows, etc).

* 1. **Herramientas y tecnologías**

**Capítulo 4**

**Resultado**

**Capítulo 5**

**Conclusiones y futuros**

* 1. **Conclusiones**
  2. **Reflexiones**
  3. **Futuros**

**Bibliografía**