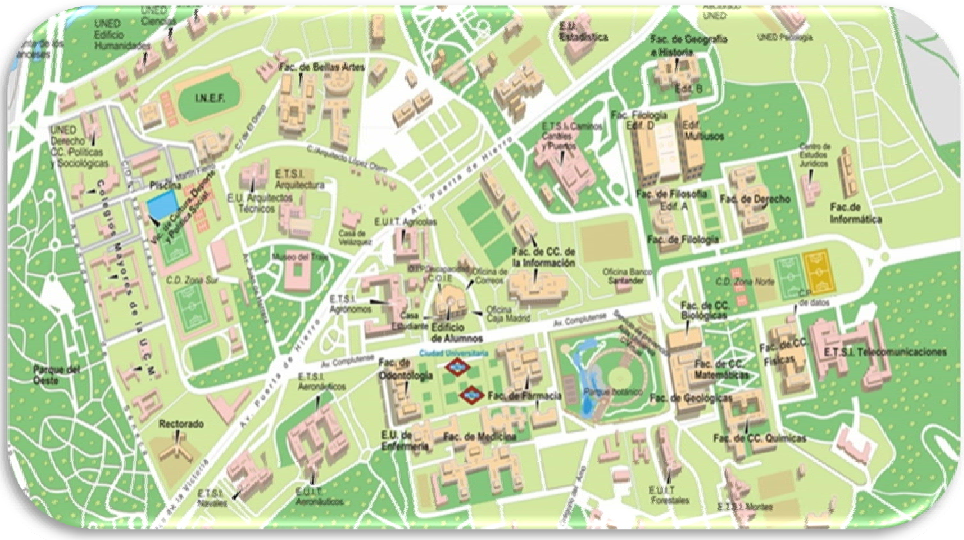
Práctica 5

Campus Ville 2.0

Fecha de entrega: 5 de junio de 2013



1. Descripción de la práctica

El juego ha pasado su primera etapa de pruebas y se han detectado algunas limitaciones, por lo que vamos a mejorar el juego antes de publicarlo.

Más concretamente, el límite de 20 jugadores y 50 edificios era adecuado para las primeras pruebas en beta, pero la empresa quiere lanzar el juego a lo grande y poder gestionar miles de usuarios y edificios, por lo que habrá que mejorar la gestión de las listas.

Por otro lado, los archivos de texto son accesibles para cualquier usuario inexperto, que podría intentar modificar los archivos y queremos evitar que esto ocurra guardando todos los datos en formato binario.

Por último, queremos modificar algunos de los algoritmos clave de la gestión de la lista ordenada añadiendo conceptos de recursión.

Por lo demás, el comportamiento y la funcionalidad de la aplicación deberían seguir siendo los mismos.

2. Guía de implementación

Desde un punto de vista técnico, los nuevos requisitos requieren varias modificaciones a desarrollar como sigue:

2.1. Uso de archivos binarios

Queremos guardar ahora ambas listas (edificios y jugadores) en archivos binarios. Los nuevos archivos se llamarán `edificios.dat` y `jugadores.dat`.

2.2. Gestión de las listas como arrays dinámicos

Para poder manejar listas más grandes, necesitamos emplear estructuras de datos mejoradas que hagan uso de la memoria dinámica para almacenar los datos. En concreto vamos a emplear **arrays dinámicos de distintos tamaños**.

Las listas de edificios y jugadores se implementarán con arrays dinámicos (Tema 9), pero optimizando la gestión de memoria: en lugar de usar un mismo array dinámico durante toda la ejecución, iremos ampliando o disminuyendo el tamaño del array para adaptarlo a las necesidades de cada momento, siguiendo esta política:

- Si el programa se inicia sin algún archivo de datos (lista en blanco), se crea el array dinámico con un tamaño de 10.
- Si ya existen los archivos de datos, se crean los arrays con una capacidad suficiente como para guardar todos los registros del correspondiente archivo, redondeado a la siguiente decena. Por ejemplo, si el archivo contiene 42 registros, se crea un array con capacidad para 50 registros.
- Si al añadir un elemento el array ya estuviese lleno, se amplía su tamaño con la siguiente fórmula[1]: nuevaCapacidad = (viejaCapacidad * 3) / 2 + 1
- No hay código para reducir el tamaño del array. Basta terminar el programa y volver a iniciarlo para que se cree el array con el tamaño adecuado.

> **Nota:** Por comodidad, se mantendrá la lista de edificios comprados igual que en la versión anterior. Observa que, dado que el array de jugadores se está creando en el montón, la lista de edificios comprados que contiene también estará en la memoria dinámica.

[1] La fórmula supone un aumento cercano al 50% de capacidad, y es empleada típicamente en muchas implementaciones avanzadas de arrays dinámicos.

2.3. Importación de archivos de texto

Dado que después de la beta tenemos ya un listado de edificios y jugadores guardado en archivos de texto, se debe añadir una nueva opción al menú del administrador:

```
9.- Importar datos de archivos de texto
```

Al seleccionar esta opción, la aplicación buscará los antiguos archivos de texto y los cargará en las listas. Cualquier dato que estuviese previamente cargado en la lista será eliminado al ejecutar esta opción.

2.4. Algoritmos recursivos

El cambio lo centraremos en la operación de búsqueda de la lista de edificios. Se deberá cambiar la función de búsqueda para implementarla como búsqueda recursiva.

3. Entrega de la práctica

La práctica se entregará a través del Campus Virtual. Se habilitará una nueva tarea **Entrega de la Practica 5** que permitirá subir un único archivo. Hay que subir un archivo comprimido llamado `GrupoXX.zip` que contenga los archivos `.h`, `.cpp` y `.dat` de la aplicación del grupo *XX*.

Fecha límite de entrega: **5 de junio a las 23:55**