Trabajo práctico 3: Implementacin en C++
Pokemon GO

Normativa

- **Límite de entrega**: Miércoles 9 de noviembre de octubre de 2016 a las 22:00 hs.
- **Límite de re-entrega**: Miércoles 14, 18 y 30 de noviembre. 22hs. de octubre de 2016 a las 12:00 hs.
- Este taller se realiza de forma **individual** y con entrega digital.
- **Normativa completa**: Para más detalles véase 'Información sobre la cursada" en el sitio Web de la materia: `http://www.dc.uba.ar/materias/aed2/2016/2c/cursada`

1. Normas de entrega adicionales

El TP contará con una entrega electrónica y, posteriormente, una entrega presencial.

Entrega electrónica

- Para ver si su trabajo pasa el conjunto de tests de la cátedra, deben enviar un mail a la dirección `algo2.dc+tp3@gmail.com` con asunto 'Grupo X' donde X es el número de grupo que les fue asignado (http://www.dc.uba.ar/materias/aed2/2016/2c/grupos).
- Al mail se deben encontrar adjuntos los archivos fuentes que representan su solución del TP. Esto es, Todos los archivos **h* y **.cpp* que representen su solución del problema (excepto los módulos de la cátedra), así como los archivos *Driver.h* y *Driver.cpp* proporcionados por la cátedra, completados correctamente.
- El sistema debería responder en un tiempo razonable si el TP pasa los test o no. Si no reciben confirmación de la recepción de su TP luego de 30 min., por favor informen su situación a la lista de docentes (algo2-doc@dc.uba.ar).
- Se pueden hacer tantas entregas como se necesiten, hasta la fecha límite de reentrega.
- Está prohibido utilizar la biblioteca estándar de C++ (STL) con la excepción de las siguientes librerías: **iostream**, **string** y **cassert**.
- Se admite hasta el estándar c++03. Está prohibido usar las funcionalidades que agrega el estándar c++11 o posteriores.

Entrega presencial

- Una vez que su trabajo pase los tests de la cátedra, deben comunicar a la lista de docentes que desean cerrar la nota mediante un pequeño coloquio grupal.
- Si alguno de los integrantes no puede asistir, con justificativo debe comunicarse con los JTPs.
- El día de la entrega deberán disponer de una copia del código que pueda ejecutarse en los laboratorios.
- Para aprobar, además del correcto funcionamiento del TP deben estar presentes todos los integrantes y responder las preguntas del corrector.

2. Enunciado

Tomando como base el diseño realizado en el TP2 y utilizando las correcciones realizadas por los docentes, se pide:

1. Implementar en C++ todo lo diseñado en el TP2, prestando atención a lo siguiente, justificando en cada caso que corresponda.

    - Respetar las interfaces de cada módulo, las funciones de la interfaz deben ser **public**.
    - Respetar estructura de representación elegida en el TP2. La misma debe ser **private**.
      Pueden adaptar la representación a la implementación de C++ por cuestiones de memoria y no por cuestiones de performance.
    - Respetar los algoritmos diseñados y sus complejidades.
      Pueden modificarlos solo ante problemas encontrados por uds. y no detectados en la corrección por el docente.

- Siempre que se pueda, corresponder los parámetos in/out de los algoritmos de diseño con parámetros por referencia y referencia constante en la implementación de C++. Usar pasaje de parámetros por copia solamente para los tipos básicos.

2. Implementar las funciones de test que crean necesarias para verificar la correctitud de la implementación (con respecto a la especificación) de cada módulo. El correcto funcionamiento de los test **no es garantía** de aprobación. Se sugiere realizar al menos un test por función y por módulo de diseño.
   Para facilitar la evaluación se provee un *driver* de test en C++ en la página web de la materia:
   http://www.dc.uba.ar/materias/aed2/2016/2c/descargas/tps/tp3
   Importante: No es requisito testear con el *driver*, pero sí es necesario que para aprobar el TP3 llenen las funciones del driver con las de sus módulos y que compile.
3. Una sugerencia para empezar es dejar la implementación de todos los métodos necesarios escrita, pero vacía, de manera de poder compilar. Pueden comentar todos los tests que requieran métodos aún no implementados de manera de poder usar la aplicación para el testing a medida que van implementando. Tengan en cuenta que algunos métodos pueden ser necesarios para muchas de las funciones de test, por lo tanto, es aconsejable empezar por esos test.
4. La implementación dada no debe perder memoria en ningún caso. Al momento de la corrección se hará el chequeo pertinente. Es una buena práctica utilizar la herramienta *valgrind* durante el desarrollo, como mínimo en la parte de testing final.
5. Se sugiere chequear las precondiciones con **assert** para facilitar la depuración de errores.
6. Para los módulos que utilicen STRING, considerar los 256 posibles caracteres del código ascii, tipo **char** en C++.
7. Pueden utilizar los módulos-cpp provistos por la cátedra los cuales implementan los módulos básicos de la materia. Los mismos no pueden ser modificados pero sí pueden ser usados como base para sus propios módulos.
   http://www.dc.uba.ar/materias/aed2/2016/2c/descargas/otros-files/modulos-cpp
8. Está prohibido utilizar la biblioteca estándar de C++ (STL) con la excepción de las siguientes librerías: **iostream**, **string** y **cassert**.