A Steroid

Paolo Cignoni
cignoni@iei.pi.cnr.it
http://vcg.iei.pi.cnr.it/~cignoni

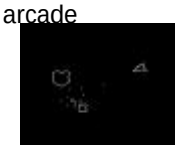
19 ottobre 1999

Introduzione

- Le prossime esercitazioni saranno tutte dedicate allo sviluppo di

  A Steroid

  (A clone of Asteroid doped by OpenGl)
- un remake della versione arcade Asteroids (1979 Atari)



A Steroid
Requisiti Fondamentali

- Portability (Win and Linux Platforms)
- Usi OpenGL
- Mantenga la stessa filosofia di gioco di Asteroids (no 3d)
- Sia piu' vario dell'originale
- Sia ben progettato (e.g. documentato, manutenibile, estendibile ecc)
- Sia un buon esempio di progetto...

Cosa Useremo

- C++ (http://www.icce.rug.nl/docs/cplusplus/)
- STL (http://www.sgi.com/Technology/STL/)
- OpenGL (http://www.opengl.org)
- GLUT (http://www.opengl.org/Documentation/GLUT.html)
- Doc++ (http://www.rsdi.com/doc++/)
- e altro....

DOC++

- It is a documentation system for C, C++ and Java generating both TeX and HTML output for online browsing of your documentation.
- The documentation is extracted directly from the C/C++ header/source files or Java class files.
- Here is a short list of highlights:
  - hierarchically structured documentation automatic class graph generation (as Java applets for HTML)
  - cross references
  - high end formatting support including typesetting of equations
- http://www.rsdi.com/doc++/

Doc++

- Doc++ legge i sorgenti e produce una documentazione in html o in TeX.
- Il meccanismo di base e' scrivere i commenti nel codice con una particolare sintassi ("/**" o "///"). Ad es.

```
/** Glut Callback on release of arrow keys.
It should stop the thrust and the rotation of the ship.
*/
void SpecialKeyUp(int key, int x, int y){
...
}
```

- La prima riga diventa il sommario del commento mentre il resto del commento diventa la descrizione

VCG::Point3<Type>

- Visual Computing Group library
  - http://vcg.iei.pi.cnr.it (mail to cignoni@iei.pi.cnr.it)
- Piccola libreria per la gestione di punti 3d con coordinate del tipo specificato a piacere
- Overloading di Tutti i classici operatori (+ - prodotto scalare, ecc.)
- Shorthand per le chiamate Opengl
  - glVertex(Point3<type>)
  - glNormal(Point3<type>)
- Typedef dei casi piu' comuni (
  - typedef Point3<float> Point3f ;
  - typedef Point3<double> Point3d ;

VCG::Point3<Type>

- Esempio:

```
using namespace vcg;
Point3f MidPoint(Point3f &p1, Point3f &p2){
        return  (p1+p2)/2.0f;
}
Point3f FaceNormal(Point3f &p1, Point3f &p2, Point3f
   &p3){
        return Normalize( (p1-p0)^(p2-p1));
}
```

STL

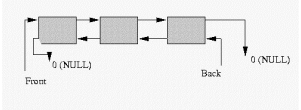
- The Standard Template Library (STL) consists of containers, generic algorithms, iterators, etc
- It is a general purpose library consisting of algorithms and data structures.
- The data structures that are used in the algorithms are abstract in the sense that the algorithms can be used on (practically) every data type.
- http://www.sgi.com/Technology/STL/

STL Vector

- The vector class implements an (expandable) array. To use the vector, the header file vector must be included:
- #include <vector>
- Vectors can be used like arrays, and can be defined with a fixed number of elements. E.g., to define a vector of 30 ints we do
- vector<int> iVector(30);
- Note the specification of the data type that is to be used: the datatype is given between angular brackets after the vector container name.

STL List

- The list class implements a list datastructure. To use the list, the header file list must be included:
- #include <list>




```
list<string>::iterator from;
for (from = target.begin(); from != target.end();
  ++from) cout << *from << " ";
cout << endl;
target.sort();
```

STL List members

- back(), returning the last element of the list.
- clear(),
- begin(), returning the first element of the list.
- empty(),
- size(),
- swap(argument), swaps two lists.

- elements may be erased:
  - erase() and clear() both erase all elements,
  - erase(pos) erases all elements starting at position pos,
  - erase(begin, end) erases elements indicated by the iterator range [begin, end).
- elements may be inserted at a certain position pos:
  - insert(pos, source) inserts source at pos,
  - insert(pos, begin, end) inserts the elements in the iterator range [begin, end).

GLUT

- GLUT is a window system independent toolkit for writing OpenGL programs.
- It implements a simple windowing application programming interface (API) for OpenGL.
- GLUT provides a portable API so you can write a single OpenGL program that works on both Win32 PCs and X11 workstation
- http://www.opengl.org/Documentation/GLUT.html

OpenGL e Glut

- Glut fornisce un'interfaccia comune con il S.O.
- Gestione interazione applicazione-utente tramite Callback
  - funzioni che sono chiamate dalla coppia S.O. GLUT in risposta a vari eventi (pressione di un tasto del mouse o della tastiera, reshape della finestra, necessita' di ridisegnare il contenuto della finestra ecc)
  - Il flusso principale dell'applicazione e' in mano al sistema operativo.

Schema Applicazione Glut

- Inizializzare
  - glutInit(&argc, argv)
- Definire e aprire una finestra
  - glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
  - glutInitWindowSize(sizex,sizey);
  - glutInitWindowPosition(xpos,ypos);
  - glutCreateWindow(char *name)
- preparare le funzione callback che SO invocherà
  - glutDisplayFunc(myRedrawFunc)
  - glutReshape(myReshapeFunc)
- Passare al SO il controllo.
  - glutMainLoop()

Gestione Animazioni e timing

- Se il controllo di quando devo disegnare lo schermo e' in mano al SO come si fa a ridisegnare continuamente lo schermo?
- Due soluzioni
  - Idle function Callback: invocata ogni volta che il so non ha nulla di piu urgente da fare
  - Timer callback: invocata ogni tot msec
- In entrambi i casi si deve sempre controllare quanto tempo e' passato dall'ultimo redraw per far sì che l'animazione sia fluida

Progetto dell'applicazione

- L'applicazione verrà sviluppata in vari passi, aggiungendo man mano sempre piu' funzionalita'
- Prima versione
  - Progettazione dei primi livelli della gerarchia delle classi principali
  - Rendering Wireframe

Esercizio 1 (**)

- Asteroide Frattale
  - Scrivere la
    class AstF : public Ast
    Il cui costruttore ha una piccola procedura che genera un profilo irregolare, memorizzandone la forma. Per generare un tale profilo partire da un quadrato o da un triangolo e aggiungere il punto di mezzo ad ogni lato e spostarlo di una piccola quantità casuale. Continuare ricorsivamente a suddividere ogni lato fino a quando non si e' raggiunto una ventina di lati.
- Nuova Astronave
- Nuova Astronave che spara 3 colpi a raggiera
- Debugging

Esercizio 2 (*)

- Nuova Astronave
  - Scrivere la
    class Ship2: public Ship
    dove la funzione di redraw e' modificata in maniera tale da disegnare (sempre in wireframe) un' astronave piu' sofisticata in cui quando il motore e' acceso la fiamma cambia forma e colore

Esercizio 3 (***)

- Nuova Astronave
  - Scrivere la
    class Ship3: public Ship
    dove (almeno) i metodi
    Bullet* Ship::Shoot()
    void GameSession::Shoot()
    sono modificati in maniera tale da permettere all'astronave di sparare tre colpi contemporanemaie a raggiera

Esercizio 4 (**)

- Debugging
- Ci sono una serie di errori (tutti involontari ma alcuni volontariamente non corretti) che vanno trovati e corretti. Ad es:
  - Posizionamento iniziale asteroidi
  - gestione reshape
  - ecc.