Khoa Công nghệ thông tin

Lớp: TH2005/02

Môn: Đồ họa máy tính

BÁO CÁO ĐỒ ÁN 3D

HỌC ĐỒ HỌA BA CHIỀU

Nhóm thực hiện:

* *0512175 Nguyễn Đăng Khoa demonuns75@yahoo.com*
* *0512186 Huỳnh Duy Khương khuong\_4kan\_nyd@yahoo.com.vn*
* *0512191 Nguyễn Trung Kiên nike17lhp@yahoo.com*

# Tên đề tài:

# Chương trình dạy học môn đồ hoạ ba chiều theo phương pháp tương tác trực quan.

# Checklist:

|  |  |  |
| --- | --- | --- |
| **Công việc** | **Đã thực hiện** | **Ghi chú** |
| 1. **YÊU CẦU BẮT BUỘC** |  |  |
| 1. **Khối hình cơ bản** |  |  |
| 1. Sphere |  |  |
| 1. Cubic |  |  |
| 1. Cylinder |  |  |
| 1. Cone |  |  |
| 1. **Các phép biến đổi hình học** |  |  |
| 1. Phép tịnh tiến |  |  |
| 1. Phép quay |  |  |
| 1. Phép đối xứng |  | Đối xứng trục |
| 1. Phép tỉ lệ |  |  |
| 1. **Các phép chiếu** |  |  |
| 1. Chiếu song song |  |  |
| 1. Chiếu phối cảnh |  |  |
| 1. **Vật liệu, chiếu sáng và tạo bóng, sương mù** |  |  |
| 1. Vật liệu |  |  |
| 1. Chiếu sáng |  |  |
| 1. Tạo bóng |  |  |
| 1. Sương mù |  |  |
| 1. **Ánh xạ texture** |  |  |
| 1. **Màu sắc** |  |  |
| 1. **YÊU CẦU NÂNG CAO** |  |  |
| 1. Khối hình cơ bản khác |  |  |
| 1. Hiệu ứng ánh sáng trên từng đối tượng |  |  |
| 1. Đọc và dựng hình từ các tập tin mô hình 3D |  |  |
| 1. Ý tưởng khác |  |  |

# Cơ sở lý thuyết:

Những kiến thức lý thuyết có liên quan đến đồ án, về OpenGL:

* Sử dụng các hàm sau để vẽ các hình khối cơ bản:
  + Mặt trước của hình được xác đinh là mặt được vẽ ngược chiều kim đồng hồ bằng cách dùng hàm glFrontFaceGL\_CCW).
  + gluCylinder: vẽ hình trụ rỗng.
  + gluDisk: vẽ mặt đĩa (sử dụng kết hợp với gluCylinder để vẽ hình nón, hình trụ đặc).
  + gluSphere: vẽ hình cầu.
  + gluQuadricOrientation: xác định hướng của hình (mặt trước, mặt sau).
  + gluNewQuadric, gluDeleteQuadric: tạo và hủy đối tượng GLUquadric dùng để xác định các thông số vẽ cần thiết khi sử dụng hàm gluCylinder, gluDisk, gluSphere.
  + glVertex3f: để xác định các điểm vẽ khi vẽ hình lập phương (vẽ từng mặt phẳng của hình).
* Sử dụng các hàm sau để thực hiện các phép biến đổi hình học:
  + Về bản chất ma trận của ModelView trong OpenGL là ma trận biến đổi Affine, nên khi vẽ một hình nào trong OpenGL thì tất cả các điểm của hình sẽ nhân với ma trận ModelView để ra tọa độ mới tương ứng với phép biến đổi Affine hiện tại, rồi mới được vẽ ra.
  + glTranslatef: (bản chất) nhân ma trận ModelView với ma trận biểu diễn phép tịnh tiến.
  + glScalef: (bản chất) nhân ma trận ModelView với ma trận biểu diễn phép co giãn.
  + glRotatef: (bản chất) nhân ma trận ModelView với ma trận biểu diễn phép quay.
  + glMultMatrixf: nhân ma trận ModelView với ma trận bất kỳ.
  + glLoadIdentity: kết hợp với hàm glMatrixMode(GL\_MODELVIEW) để ma trận ModelView thành ma trận đơn vị.
* Sử dụng các hàm sau để thực hiện các phép chiếu:
  + glOrtho: thiết đặt khung nhìn của phép chiếu trực giao. Khung nhìn của phép chiếu trực giao là một hình hộp.
  + gluPerspective: thiết đặt khung nhìn của phép chiếu phối cảnh. Khung nhìn của phép chiếu phối cảnh là một chóp cụt.
* Sử dụng các hàm sau để thiết lập vật liệu cho hình khối, chiếu sáng, tạo bóng, sương mù:
  + glEnable(GL\_COLOR\_MATERIAL), glDisable(GL\_COLOR\_MATERIAL): bật/tắt hiệu ứng vật liệu cho các hình khối.
  + glColorMaterial, glMaterialfv: thiết đặt các thông số liên quan của vật liệu như độ sáng, màu vật liệu phụ thuộc vào yếu tố môi trường nào.
  + glEnable(GL\_LIGHTING), glDisable(GL\_LIGHTING): bật/tắt hiệu ứng chiếu sáng.
  + glLightfv: hiệu chỉnh các thông số của một nguồn sáng trong OpenGL (có 8 nguồn sáng trong OpenGL) như: vị trí, hướng chiếu, góc chiếu, màu, … của nguồn sáng.
  + glNormal3f: chỉ định vector pháp tuyến của mặt phẳng, dùng trong kỹ thuật chiếu sáng.
  + Tạo bóng: sử dụng một phép biến đổi để biến đổi các hình 3D thành dạng phẳng và nằm trên một mặt phằng xác định, và các hình được vẽ ra là màu đen.
  + glEnable(GL\_FOG), glDisable(GL\_FOG): bật/tắt hiệu ứng sương mù.
  + glFogfv, glFogf: thiết đặt các thông số liên quan đến hiệu ứng sương mù như: màu sắc, vị trí bắt đầu, vị trí kết thúc, độ dày đặc, … của sương mù.
* Sử dụng các hàm sau để ánh xạ texture:
  + glGenTextures: tạo texture.
  + glBindTexture: áp dụng texture cho các hình khối.
  + glTexImage2D: đưa dữ liệu (ảnh) vào texture.
  + glTexParameteri: thay đổi các tham số của texture như bộ lọc… của texture.
  + gluQuadricTexture: cho phép áp dụng texture nào cho hình. Dùng kết hợp với các hàm gluSphere, gluCylinder, gluDisk để ánh xạ hiệu ứng texture.
  + glTexCoord2f: ánh xạ texture cho các hình dạng phẳng.
* Sử dụng các hàm để xác định màu sắc cho hình khối:
  + glColor4f: xác định màu cho các hình.

# Ý tưởng cài đặt:

* Cấu trúc của chương trình được thiết kế theo hướng đối tượng, gồm có các lớp chính sau:



* Danh sách các thuộc tính, phương thức của từng lớp:
* \_ObjectOpenGL: lớp trừu tượng, cung cấp các phương thức thuần ảo để các lớp con định nghĩa lại, các phương thức dùng chung cho các lớp con. Là lớp mô tả các thuộc tính chung của hình khối.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mMatrix | float[] | Ma trận (mảng 16 phần tử để mô tả ma trận 4x4) mô tả phép biến đổi Affine hiện tại của hình. Được sử dụng trong các hàm Rotate, Scale, Translate, Axisymmetric. |
|  | mCenter | float[] | Tọa độ tâm của hình (mảng 3 phần tử ứng với x, y, z). Tọa độ tâm được qui đinh tùy theo từng hình, không phải lúc nào cũng nằm ở tâm của hình. Ví dụ: hình nón thì mCenter là tâm của mặt phẳng đáy. |
|  | mColor | int[] | Màu của hình. Mảng 4 phần tử ứng với R, G, B, A. |
|  | mShininess | int | Độ sáng của vật liệu (0..128). |
|  | mEnable | Boolean | Bật tắt chế độ vẽ hình ra màn hình. |
|  | mEnableTexture | Boolean | Bật tắt ánh xạ texture lên hình. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | Rotate | void | Hàm ảo. Quay hình một góc Angle, quay một trục bất kỳ (trục có thể có một đầu đặt tại tâm tọa độ hoặc không). |
|  | Translate | void | Hàm ảo. Tịnh tiến hình theo ba trục Ox, Oy, Oz. |
|  | Scale | void | Hàm ảo. Co giãn hình theo 3 trục Ox, Oy, Oz. |
|  | Axisymmetric | void | Hàm ảo. Thực hiện phép đối xứng với một trục bất kỳ. |
|  | SetColor | void | Đặt lại màu của hình. |
|  | Draw | void | Hàm ảo. Vẽ hình ra màn hình. Tại lớp này, chỉ đặt lại màu và vật liệu cho hình. Các lớp sau sẽ thực hiện vẽ hình tương ứng với lớp đó. |

* \_Cone: kế thừa từ lớp \_ObjectOpenGL. Là lớp thể hiện thông tin và phụ trách vẽ hình nón.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mBase | double | Bán kính đáy của hình nón. |
|  | mHeight | double | Chiều cao của hình nón. |
|  | mSlices | int | Số lát dọc dùng để vẽ hình nón. Là 1 tham số trong hàm gluCylinder, gluDisk của OpenGL |
|  | mStacks | int | Số lát ngang dùng để vẽ hình nón. Là 1 tham số trong hàm gluCylinder, gluDisk của OpenGL |
|  | QuadricCone | GLUquadric | Là 1 tham số trong hàm gluCylinder. |
|  | QuadricDisk | GLUquadric | Là 1 tham số trong hàm gluDisk. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| Không có phương thức mới | | | |

* \_Cubic: kế thừa từ lớp \_ObjectOpenGL. Là lớp thể hiện thông tin và phụ trách vẽ hình lập phương.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mSide | double | Chiều dài một cạnh của hình lập phương |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| Không có phương thức mới | | | |

* \_Cylinder: kế thừa từ lớp \_ObjectOpenGL. Là lớp thể hiện thông tin và phụ trách vẽ hình trụ.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mBase | double | Bán kính 2 đáy của hình trụ. |
|  | mHeight | double | Chiều cao của hình trụ. |
|  | mSlices | int | Số lát dọc dùng để vẽ hình trụ. Là 1 tham số trong hàm gluCylinder, gluDisk của OpenGL |
|  | mStacks | int | Số lát ngang dùng để vẽ hình trụ. Là 1 tham số trong hàm gluCylinder, gluDisk của OpenGL |
|  | mQuadricCylinder | GLUquadric | Là 1 tham số trong hàm gluCylinder. |
|  | QuadricDisk1 | GLUquadric | Là 1 tham số trong hàm gluDisk. |
|  | QuadricDisk2 | GLUquadric | Là 1 tham số trong hàm gluDisk. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| Không có phương thức mới | | | |

* \_Sphere : kế thừa từ lớp \_ObjectOpenGL. Là lớp thể hiện thông tin và phụ trách vẽ hình cầu.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mBase | double | Bán kính của hình cầu. |
|  | mSlices | int | Số lát dọc dùng để vẽ hình cầu. Là 1 tham số trong hàm gluSphere của OpenGL |
|  | mStacks | int | Số lát ngang dùng để vẽ hình cầu. Là 1 tham số trong hàm gluSphere của OpenGL |
|  | mQuadric | GLUquadric | Là 1 tham số trong hàm gluSphere. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| Không có phương thức mới | | | |

* \_LightSource: kế thừa từ lớp \_ObjectOpenGL. Là lớp thể hiện thông tin và phụ trách vẽ hình nguồn sáng.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mLightPos | float[] | Vị trí của nguồn sáng. Mảng 3 chiều ứng với tọa độ x, y, z |
|  | mSpotDir | float[] | Hướng của nguồn sáng. Mảng 3 chiều ứng với tọa độ x, y, z, là tọa độ của một điểm, điểm này và vị trí của nguồn sáng sẽ tạo thành hướng của nguồn sáng. |
|  | mSpotCutOff | float | Góc chiếu sáng của nguồn sáng. |
|  | mAmbientLight | float[] | Màu của ánh sáng xung quanh. Mảng gồm 4 phần tử ứng với R, G, B, A. Còn màu phản chiếu của nguồn sáng là mColor đặt ở lớp \_ObjectOpenGL. |
|  | mIndex | int | Cho biết sẽ bật nguồn sáng nào trong OpenGL. |
|  | mQuadricCone | GLUquadric | Được dùng để vẽ nguồn sáng, chủ yếu là để minh họa nguồn sáng đang nằm ở đâu. Là tham số của hàm gluCylinder trong OpenGL. |
|  | mQuadricSphere | GLUquadric | Được dùng để vẽ nguồn sáng. Là tham số của hàm gluSphere trong OpenGL. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| Không có phương thức mới | | | |

* \_Vector: kế thừa từ lớp \_ObjectOpenGL, sử dụng lớp \_Vector3D. Là lớp thể hiện thông tin và phụ trách vẽ vector.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mBegin | \_Vector3D | Điểm bắt đầu của vector. |
|  | mEnd | \_Vector3D | Điểm kết thúc của vector. |
|  | pObj | GLUquadric | Được dùng để vẽ đầu mũi tên của vector. Là tham số của hàm gluCylinder, gluDisk trong OpenGL. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
| Không có phương thức mới | | | |

* \_UnitAxes: là lớp phụ trách vẽ trục tọa độ.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mRange | float | Cho biết độ dài của các trục. |
|  | mEnable | Boolean | Bật tắt việc vẽ trục tọa độ ra màn hình. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | Draw | void | Vẽ trục tọa độ ra màn hình. |

* \_Ground: là lớp phụ trách vẽ nền ra màn hình, dùng trong đổ bóng.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mvPlaneEquation | float[] | Chứa các hệ số của phương trình mặt phẳng. Mảng 4 phần tử ứng với hệ số A, B, C, D trong phương trình đường thẳng. |
|  | mRange | float | Hệ số dùng để vẽ mặt đất. mRange có giá trị bằng với giá trị của trục Ox. |
|  | mEnable | Boolean | Bật tắt việc vẽ nền ra màn hình. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | Draw | void | Vẽ nền ra màn hình. Nền nằm trên mặt phẳng Oxz. |

* \_Shadow: sử dụng lớp \_Ground. Lớp dùng để vẽ bóng.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mShadowMat | float[] | Ma trận để biến đổi các hình 3D thành dạng phẳng và nằm trên mặt nền. Ma trận này thay đổi khi nguồn sáng thay đổi vị trí. |
|  | mGround | \_Ground | Tham chiếu đến đổi tượng nền. Dùng để tính ma trận biến đổi. |
|  | mEnable | Boolean | Bật tắt việc vẽ bóng ra màn hình. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | CalculatorMatrix | void | Tính lại ma trận biến đổi ứng với vị trí mới của nguồn sáng. |
|  | Draw | void | Vẽ bóng của các hình (tham số của hàm là tập hợp các hình cần tạo bóng) lên nền. |

* \_Texture: Lớp dùng để điều khiển việc ánh xạ texture lên các hình.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mTexture | int[] | Cấu trúc dùng để lưu trữ texture cần ánh xạ. Mảng 1 phần tử. |
|  | mEnable | Boolean | Bật tắt việc ánh xạ texture lên hình. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | LoadTexture | Boolean | Nạp texture lên bộ nhớ. Dùng hình để làm texture. Tham sô của hàm là đường dẫn chỉ đến file ảnh. Trả về false nếu nạp texture thất bại. |
|  | Apply | void | Cho phép áp dụng texture lên các hình. (Áp dụng được hay không còn tùy thuộc vào biến mEnable). |

* \_Fog: Lớp dùng để điều khiển việc tạo sương mù.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mColor | int[] | Màu của sương mù. Mảng 4 phần tử ứng với R, G, B, A. |
|  | mStart | float | Nơi bắt đầu có sương mù. (Tính theo trục Oz). |
|  | mEnd | float | Nơi kết thúc của sương mù. (Tính theo trục Oz). |
|  | mDensity | int | Độ dày đặc của sương mù. |
|  | mEnable | Boolean | Bật tắt việc tạo sương mù. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | Apply | void | Cho phép tạo sương mù. (Áp dụng được hay không còn tùy thuộc vào biến mEnable). |

* \_Vector3D: Lớp mô tả thông tin và các phương thức sử lý của một vector trong không gian Oxyz.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mX | float | Giá trị x của vector. |
|  | mY | float | Giá trị y của vector. |
|  | mZ | float | Giá trị z của vector. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | operator - | \_Vector3D | Thực hiện phép trừ 2 vector. |
|  | operator \_Vector3D(float[] P) | \_Vector3D | Ép kiểu float[] sang kiểu \_Vector3D. |
|  | operator float[](\_Vector3D P) | \_Vector3D | Ép kiểu \_Vector3D sang kiểu float[]. |

* \_Affine: Lớp mô tả thông tin về một phép biến đổi Affine bất kỳ.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | mMatrix | float[] | Ma trận biến đổi của phép biến đổi Affine. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | Rotate | void | Chuyển mMatrix thành ma trận mô tả cho phép quay quanh một trục bất kỳ với góc quay Angle. |
|  | Scale | void | Chuyển mMatrix thành ma trận mô tả cho phép co giãn theo theo 3 trục Ox, Oy, Oz. |
|  | Translate | void | Chuyển mMatrix thành ma trận mô tả cho phép tịnh tiến theo 3 trục Ox, Oy, Oz. |
|  | Axisymmetric | void | Chuyển mMatrix thành ma trận mô tả cho phép đối xứng qua một trục bất kỳ. |
|  | Identity | void | Chuyển mMatrix thành ma trận đơn vị. |
|  | ConvertPoint | float[] | Áp dụng phép biến đổi Affine lên điểm và trả về tọa độ mới ứng với phép biến đổi đó. |
|  | operator+ | \_Affine | Kết hợp 2 phép biến đổi Affine. |

* \_Math3D: Lớp chứa các hàm static, các hàm này thực hiện các phép toán 3 chiều. Tất cả các hàm này đều được lấy từ thư viện gltool.h

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
| Không có phương thức mới | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | GetVectorLengthSquared | float | Lấy bình phương chiều dài của đoạn thẳng. |
|  | ScaleVector3 | void | Thực hiện co giãn vector theo hệ số cho trước. |
|  | GetVectorLength | float | Lấy chiều dài của vector. |
|  | NormalizeVector | void | Chuyển một vector thành vector có chiều dài đơn vị. |
|  | CrossProduct | void | Tích có hướng của 2 vector. |
|  | GetPlaneEquation | void | Tìm phương trình mặt phẳng. |
|  | MakePlanarShadowMatrix | void | Lập ma trận biến đổi làm các hình 3 chiều thành hình phẳng và nằm trên một mặt phẳng dựa vào phương trình mặt phẳng và vị trí đèn. |
|  | GetDistance | float | Tính chiều dài của đoạn thẳng. |
|  | GetAngleBetweenVectors | float | Tính góc giữa 2 vector. |
|  | FindNormalVector | float[] | Tìm vector pháp tuyến của mặt phẳng. |
|  | CopyMatrix | void | Sao chép các giá trị từ ma trận nguồn sang ma trận đích. |

* \_Form1: lớp form, dùng để xử lý giao diện, Trong lớp này chứa nhiều control cơ bản, nên sẽ không liêt kê các phương thức và thuộc tính liên quan đến control. Chỉ nêu các thuộc tính và hàm có liên quan đến các lớp ở trên.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Ghi chú |
|  | lstObj | List<\_ObjectOpenGL> | Tập hợp các hình khối cơ bản. |
|  | lstLightSource | List<\_LightSource> | Tập hợp các nguồn sáng |
|  | lstVector | List<\_Vector> | Tập hợp các vector, chỉ dùng để minh họa trục khi thực hiện phép quay quanh trục, hoặc phép đối xứng trục. |
|  | fog | \_Fog | Đối tượng dùng để lưu các thông số cần thiết khi tạo sương mù. |
|  | texture | \_Texture | Đối tượng dùng để lưu các thông số cần thiết khi ánh xạ texture. |
|  | unitaxes | \_UnitAxes | Đối tượng dùng để vẽ trục tọa độ. |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Phương thức | Kiểu trả về | Ghi chú |
|  | GL\_Paint | void | Hàm chính, dùng để vẽ các hình lên màn hình, trục tọa độ, tạo bóng, sương mù, texture, nền, nguồn sáng. |
|  | SetupRC | void | Thiết lập các thông số ban đầu cho OpenGL. |
|  | SetView | void | Thiết đặt phép chiếu trực giao hoặc phép chiếu phối cảnh. |
|  | … | … | … |

* Cách chương trình thực hiện:
  + Để chạy được chương trình phải có bộ Tao Framework, download tại <http://superb-east.dl.sourceforge.net/sourceforge/taoframework/tao-2.0.0-setup.exe> (13.51MB).
  + Lớp Form1 có một đối tượng simpleOpenGlControl1 thuộc lớp simpleOpenGlControl (thuộc Tao Framework), tất cả các hàm OpenGL sẽ được vẽ lên trên đối tượng này.
  + Khi người dùng yêu cầu một chức năng nào đó trên giao diện thì chương trình chỉ việc gọi chức năng tương ứng được cung cấp bởi các lớp đã trình bày, rồi sau đó cho đối tượng simpleOpenGlControl vẽ lại là được.

# Hướng dẫn sử dụng:

Chưa làm kịp.

# Tự đánh giá:

* Ưu điểm:
  + Về phần cài đặt: chương trình được tổ chức theo hướng đối tượng, nên khi thiết kế giao diện chỉ cần dùng hàm của những lớp bên dưới để thực hiện, nên dễ tổ chức và quản lý.
  + Về chương trình: dễ sử dụng, cho phép chỉnh nhiều thông số.
* Khuyết điểm:
  + Ít hình khối.
  + Chỉ cho phép vẽ một hình trên màn hình nên không thấy được tính tương tác.
  + Về phần tạo bóng vẫn chưa hoàn thiện, khi hình và nguồn sáng nằm về 2 phía của nền thì vẫn hiện bóng (sai).
* Các ý tưởng chưa cài đặt kịp:
  + Cho phép vẽ nhiều hình trên màn hình, về phần thiết kế bên dưới thì đã cho phép vẽ nhiều hình trên màn hình (biến lstObj có kiểu list).
  + Cho phép 2 hình tương tác với nhau với nhau.
  + Cho phép thực hiện Animation, ví dụ người dùng tịnh tiến hình thì cho hình di chuyển từ từ đến vị trí mới.
  + Cho phép người dùng nhập vào nhiều phép biến đổi liên tiếp rồi dùng hiệu ứng Animation để mô tả các phép biến đổi đó.
* Nhận xét về đề bài project:
  + - Tương đối khó.
    - Phù hợp với môn đồ họa máy tính.
    - Nên cho đồ án vào đầu mỗi học kỳ để sinh viên có thời gian chuẩn bị kỹ hơn.

# Tài liệu tham khảo:

* OpenGL® SUPERBIBLE Fourth Edition Comprehensive Tutorial and Reference; Richard S. Wright, Jr., Benjamin Lipchak, Nicholas Haemel
* <http://nehe.gamedev.net/>
* <http://members.hellug.gr/nkour/Tao.OpenGL_Builder/Tao_Builder-html.zip>
* Slide của thầy Bùi Tiến Lên.