Motor

Como já foi referido no diagrama de classes, o **motor** contém um apontador para uma estrutura **Grupo** (correspondente à **cena**) e **Camera**.

Quando se inicia (este quando se inicia, não parece ficar muito bem, acho que temos de tentar arranjar outra forma de começar este parágrafo), faz o *parsing* do ficheiro passado como argumento, com o auxílio do **parser**, carregando para a **cena** o apontador para **Grupo** devolvido pela função de *parsing*. Depois, sempre que é chamada a função **renderScene** este chama uma função **imprimeGrupo** que tem como finalidade apresentar as imagens correspondentes ao grupo passado como argumento (com as devidas transformações).

A função **imprimeGrupo** percorre as transformações geométricas do mesmo, aplicando a operação para cada uma. Depois percorre as **Figuras** que o grupo contém apresentando-as com o auxílio da função **imprimeFigura**. (Após isto,) Depois percorre os **grupos** filhos presentes no mesmo e aplica a mesma função a esses **grupos**. Para que as transformações geométricas sejam realizadas a todo o grupo (modelos e grupos) e apenas a esse grupo, no início é realizada a operação *glPushMatrix()* e no final *glPopMatrix()*.

void imprimeGrupo(Grupo\* g){

glPushMatrix();

vector<Operacao\*> operacoes = g->getOperacoes();

for(Operacao op: operacoes){

op->aplicaOperacao();

}

vector<Figura\*> figuras = g->getFiguras();

imprimeFigura(figuras);

vector<Grupo\*> gruposFilhos = g->getGrupos();

for(Grupo\* filho: gruposFilhos){

imprimeGrupo(filho);

}

glPopMatrix();

}

A função **imprimeFigura** recebe como argumento um apontador para uma **Figura** e percorre cada ponto dessa **figura**, desenhando-o cada ponto (desenhando-os) (a ordem é importante) com a função *glVertex3f*.

void imprimeFigura(vector<Figura\*> figuras){

for(Figura\* f: figuras){

vector<Ponto\*> pontos = f->getPontos();

for (Ponto \*p: pontos) {

glVertex3d(p->getX(), p->getY(), p->getZ());

}

}

glEnd();

}

POR PARTE DE EXPLICAÇÃO DA CÂMERA!