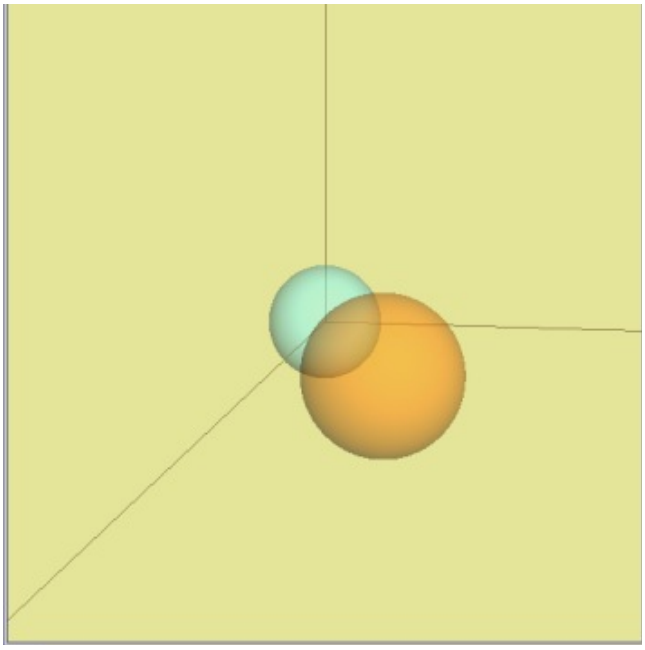
Trabalho **5**

Transparências



Computação Gráfica

Aulas práticas

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

Jorge Henriques/Paulo de Carvalho

Abril de 2011

Índice

Trabalho para Avaliação

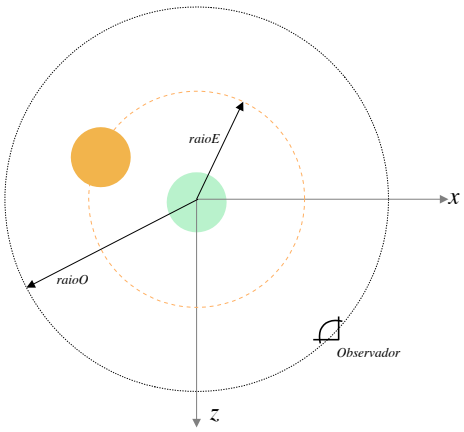
Período de Realização:	1 semanas
Data máxima para entrega:	Alunos da TP1: 17 de Maio (24h) Alunos da Tp2-TP7:19 de Maio (24h)
Local e formato de Entrega:	Cacifo do Prof. Paulo de Carvalho, em **CD** com o código fonte e o projecto compilado (não necessita de relatório).
Defesa:	obrigatória; a defesa será realizada nas aulas PL; inscrição obrigatória

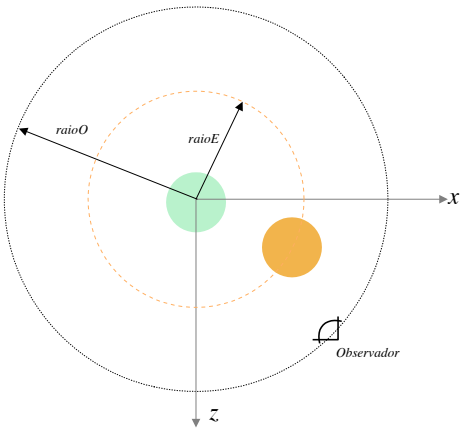
1. Introdução

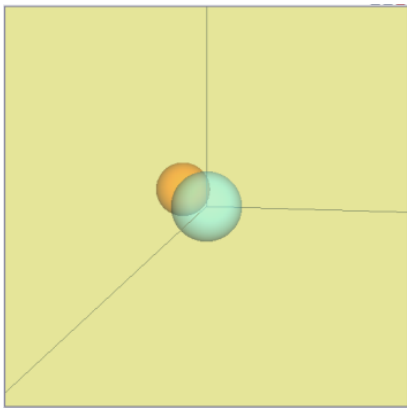
Pretende-se implementar em OpenGL um programa que permita **visualizar duas esferas transparentes**.

Na figura seguinte mostra-se duas situações do resultado pretendido.
Na primeira situação a esfera verde encontra-se localizada entre o observador e a esfera laranja, na segunda a esfera laranja situa-se entre o observador e a esfera verde.

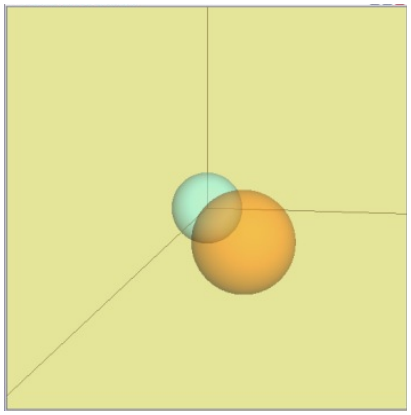
Em qualquer dos casos as esferas devem ser transparentes para o observador.







Situação 1



Situação 2

2. *Transparência em OpenGL*

Para exibir objectos transparentes em OpenGL, utilizam-se as funções de **Blend**. Estas funções combinam a cor dos objectos já desenhados (que constituem o ecrã actual) com a cor do objecto que está a ser desenhado num dado momento.

Activação

Em primeiro lugar é necessário activar as funções de BLEND. Para o efeito, usa-se:

```
glEnable(GL_BLEND);
```

Quando for necessário desenhar um objecto que não é transparente, deve-se desactivar o BLEND:

```
glDisable(GL_BLEND);
```

Definição do nível de Transparência de um Objecto

Para definir o nível de transparência de um objecto é necessário especificar a componente ALFA da sua cor. Para tal usa-se o quarto parâmetro na função glColor.

```
glColor4f( r, g, b, alfa);
```

Este quarto parâmetro **alfa** pode variar entre 0 e 1, sendo que 0 identifica um objecto totalmente transparente e 1 um objecto opaco.

Deve ser activado o "teste de profundidade", z-buffer.

```
glEnable(GL_DEPTH_TEST)
```

Definição das Funções de Blend (combinação)

As funções de BLEND definem como é feita a combinação entre as cores do objecto que está a ser desenhado e as imagens previamente desenhadas.
Esta combinação é feita a partir de uma média ponderada entre os pixeis do objecto a desenhar (**source**) e os já existentes (**destination**). A função OpenGl que permite definir este pesos é **glBlendFunc**.

```
void glBlendFunc(GLenum sfactor, GLenum dfactor).
```

Esta função recebe dois parâmetros: o primeiro define o "peso" da cor dos novos pixeis e o segundo o "peso" da cor dos pixeis já existentes. Estes pesos, em OpenGL, são função do nível de transparência do objecto, ou seja, do valor do seu alfa.

Por exemplo, a chamada:

```
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
```

define a cor dos novos pixeis como:

novaCor = corSource * alfaSource + corDestination* (1-alfaSource);

ou

novaCor = corNovoObjeto*alfaNovoObjeto + corExistente* (1-alfaNovoObjeto);

Neste caso, as constantes tem o seguinte significado:

- GL_SRC_ALPHA: define para o peso da cor do objecto a desenhar o valor do alfa da sua cor;
- GL_ONE_MINUS_SRC_ALPHA: define para o peso da cor que já está desenhado no ecran é de (1 - alfa), onde alfa é o nível de transparência do objecto que está a ser desenhado.

Algumas regras

1.

Antes de mais, os objectos transparentes devem ser desenhados depois dos opacos.

2.

Nem todas as combinações de factores de fonte e de destino fazem sentido (glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

3.

Uma forma de desenhar uma figura composta por uma mistura igual de duas imagens, é:

1. Atribuir ao factor source o valor GL_ONE e ao factor destino GL_ZERO. Desenhar o primeiro objecto (opaca, que funciona como a source);

2. Depois atribuir GL_SRC_ALPHA ao factor source e GL_ONE_MINUS_SRC_ALPHA ao factor destino. Considerar alfa do segundo objecto desenhar =0.5 (igual nivel de transparência).
Desenhar depois o segundo objecto (transparente, que funciona agora como source e a opaca como destination).

```
CorObjectoOpaco={ r, g, b, x};
CorObjectoTransparente={ r,g,b, 0.5};

glBlendFunc (GL_ONE, GL_ZERO);
desenhaObjectoOpaco();

glBlendFunc (GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
desenhaObjectoTransparente();
```

3. *Trabalho*

Entrega

O trabalho é para entregar

A fazer

Implemente um programa que desenhe duas esferas nas condições descritas no enunciado.

O seu programa deverá permitir variar:

- o nível de transparência (tecla T para aumentar e tecla R para diminuir)
- a velocidade de rotação da esfera (tecla A para aumentar e tecla S para diminuir)