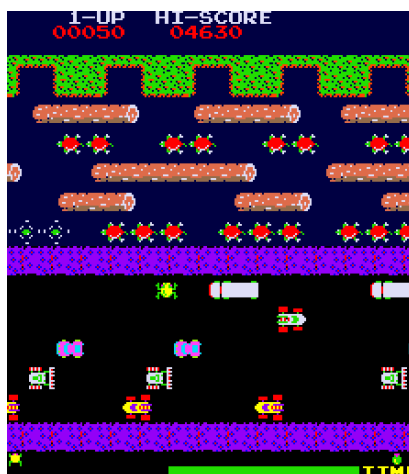


Parte 1

Cena Simples Interativa com Câmara Fixa

Breve Introdução ao Tema

O jogo *Frogger* foi criado em 1981 numa parceria entre a SEGA e a Konami, para consolas *arcade*. O jogador controla um sapo que tem de atravessar, em tempo limitado, um rio e uma estrada cheios de perigos. Na estrada o sapo deve evitar carros, camiões, autocarros e até ciclistas, enquanto no rio deve usar as costas dos crocodilos, e das tartarugas, assim como os troncos para cruzar o rio sem cair à água. Em ambos os casos, os objectos deslocam-se horizontalmente. O sapo tem de sair da parte inferior do ecrã e atingir uma das “casas de sapo” que existem na parte superior. Sempre que é atingido por um dos obstáculos na estrada ou no rio, o sapo morre, i.e. o jogador perde uma das cinco vidas.



Nas últimas décadas têm aparecido inúmeras versões deste jogo, incluindo algumas reimplementações com o grafismo original, ilustrado na figura acima, juntamente com a capa do jogo. Algumas destas versões podem ser jogadas online¹.

¹ <http://www.happyhopper.org/>

Trabalho a realizar

O objectivo dos trabalhos de laboratório de Computação Gráfica deste ano é recriar este clássico numa versão 3D, recorrendo a **C++** e **OpenGL**. A ideia é manter a jogabilidade original alterando a perspectiva gráfica para que os vários elementos do jogo tenham um aspecto 3D. Podem ver um exemplo para inspiração na figura seguinte.



O trabalho está dividido em quatro partes que serão avaliadas individualmente ao longo do semestre. Em cada uma destas avaliações existem objectivos e tarefas específicas para que possam explorar as várias componentes do programa de Computação Gráfica.

O resto deste documento refere-se à primeira parte do trabalho. As outras três partes serão publicadas ao longo do semestre.

Objectivos

Os objectivos da primeira parte dos trabalhos de laboratório são compreender e implementar a arquitectura de uma aplicação gráfica interativa e explorar os conceitos básicos de modelação.

A avaliação da primeira parte do trabalho será realizada na semana de **13 a 17 de Outubro** e corresponde a **6 valores** da nota do laboratório. A realização deste trabalho tem um esforço estimado de **14 horas** por elemento do grupo, distribuído por três semanas.

Tarefas

As tarefas para a primeira parte são:

1. Modelar o rio e as suas margens, assim como a estrada e as suas bermas usando um cubo para cada elemento. Definir uma câmara fixa com uma vista de topo sobre a cena utilizando uma projecção ortogonal (semelhante à vista 2D do jogo original). [1,5 valores]
2. Modelar o sapo, os carros e os troncos, recorrendo a objectos geométricos tridimensionais simples (cubos e esferas). O sapo deve ser composto por mais do que um objecto geométrico, num mínimo de dois objectos, enquanto o carro deve ser composto por um mínimo de três. [2,5 valores]
3. Controlar o movimento do sapo com o teclado utilizando as teclas 'O' para mover o sapo para a esquerda, 'P' para mover para a direita, 'Q' para mover para cima e 'A' para mover para baixo. O sapo deve iniciar o movimento quando a tecla é pressionada e parar quando a tecla é largada. [2,0 valores]

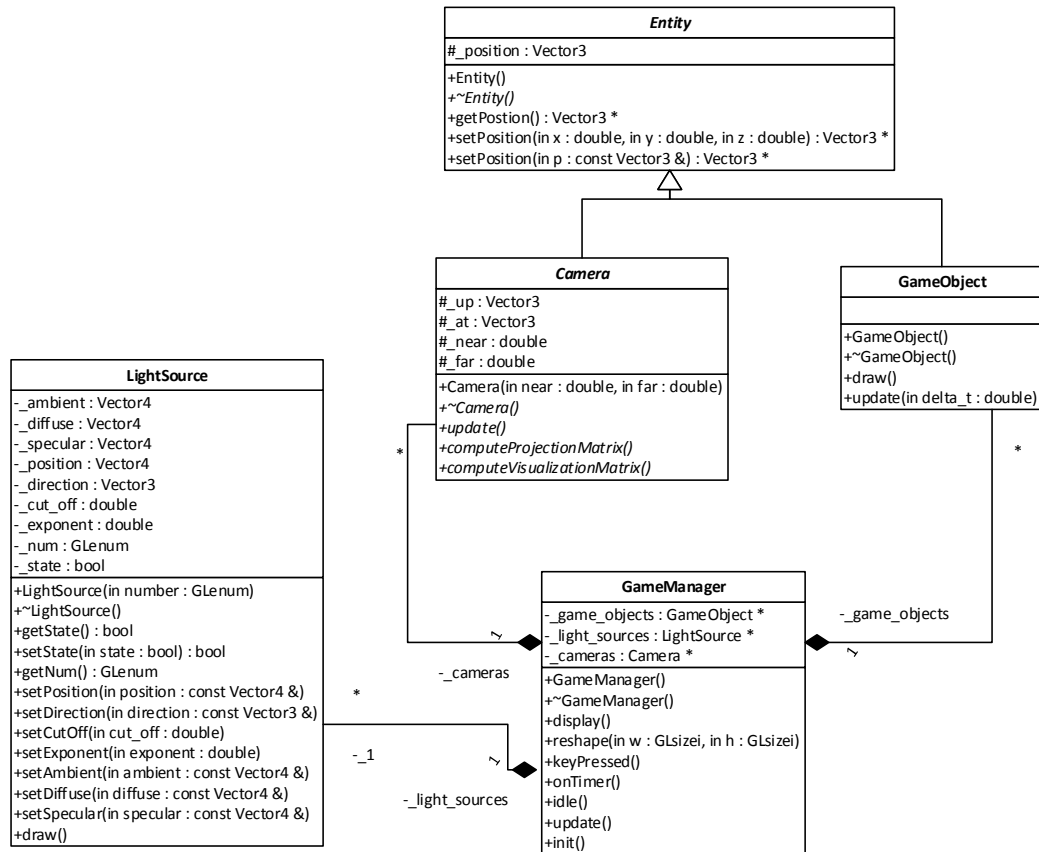
Sugestões

1. Usar programação orientada a objectos, com a estrutura de classes apresentada nos diagramas UML em anexo. Siga sempre boas práticas de programação que permitam a reutilização do código em entregas posteriores e facilitem a escalabilidade futura.
2. Desenhar os objectos em papel antes de escrever o código *OpenGL* ajuda muito a perceber que primitivas e transformações devem ser aplicadas. No anexo B pode encontrar **um exemplo** de um esboço em papel para um objecto (**não usar** este esboço).
3. As rodas do carro podem ser criadas recorrendo a *torus*.
4. Algumas das funções a estudar:
 - `glViewport`, `glMatrixMode`, `glOrtho`
 - `glTranslate`, `glRotate`, `glScale`
 - `glPushMatrix`, `glPopMatrix`
 - `glutSolidCube`, `glutSolidSphere`
 - `glutKeyboardFunc`
 - `glutTimerFunc`
 - `glutGet(GLUT_ELAPSED_TIME)`

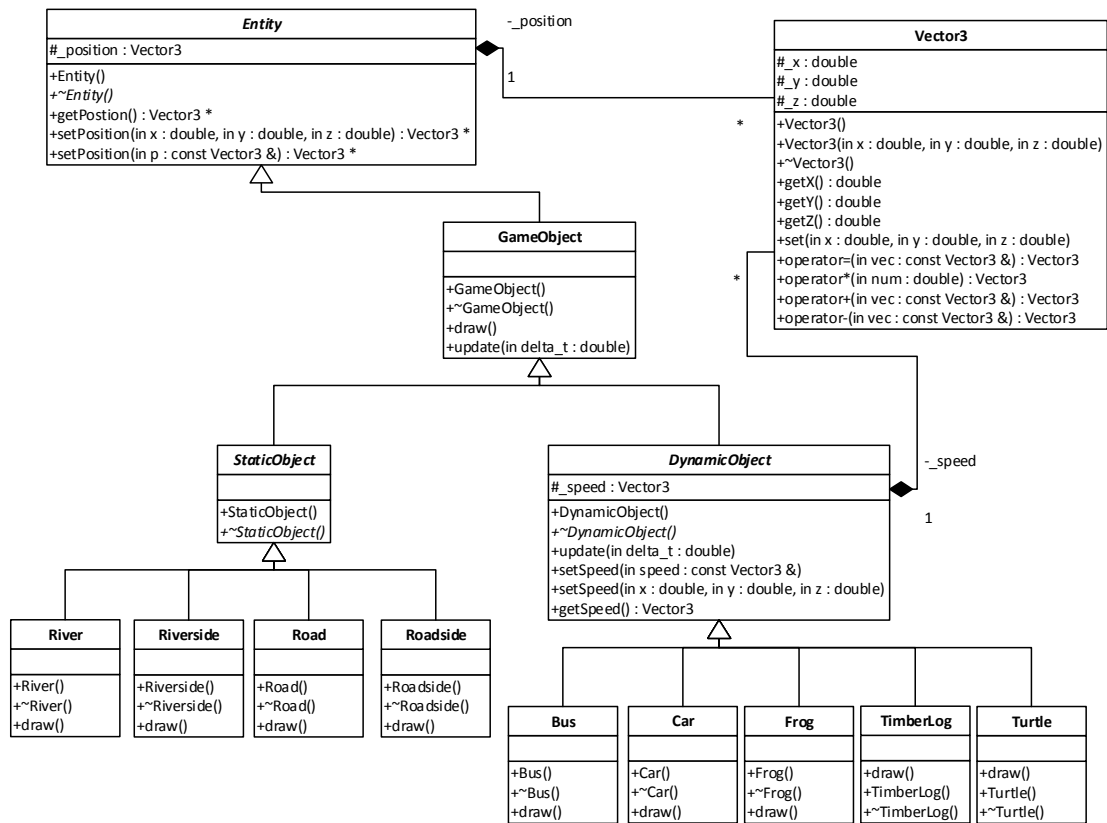
Anexo A

Diagramas UML

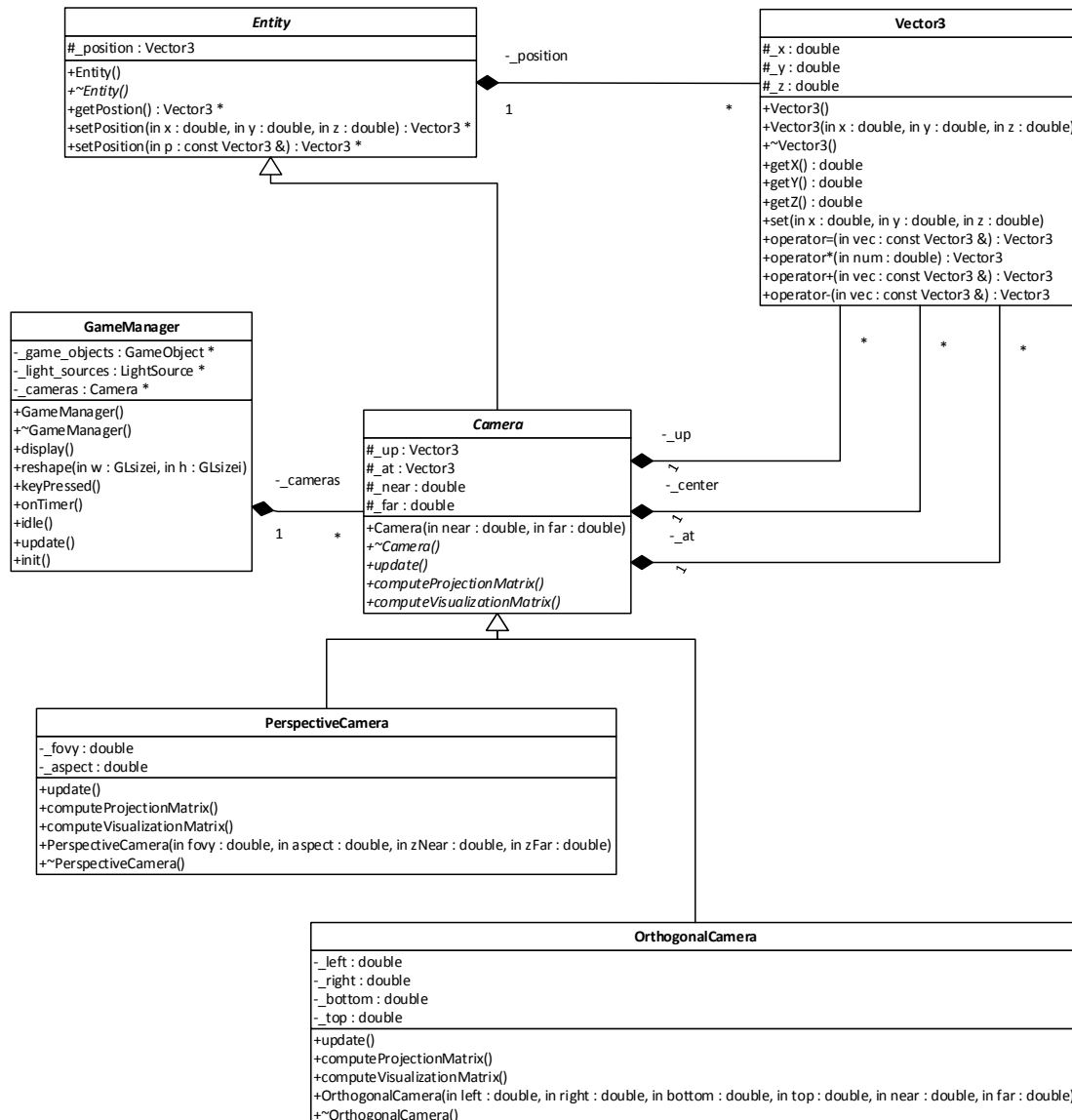
A1. Gestor de Jogo e classes associadas



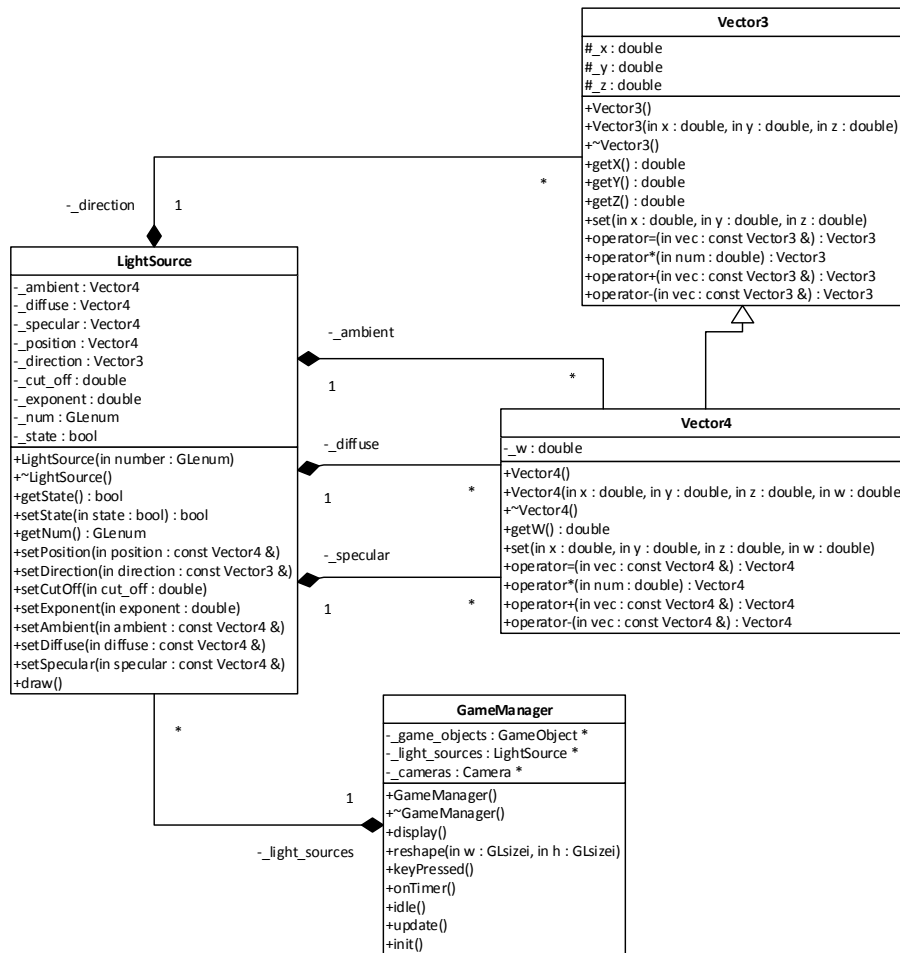
A2. Objecto do Jogo e classes associadas



A3. Câmara Virtual e classes associadas

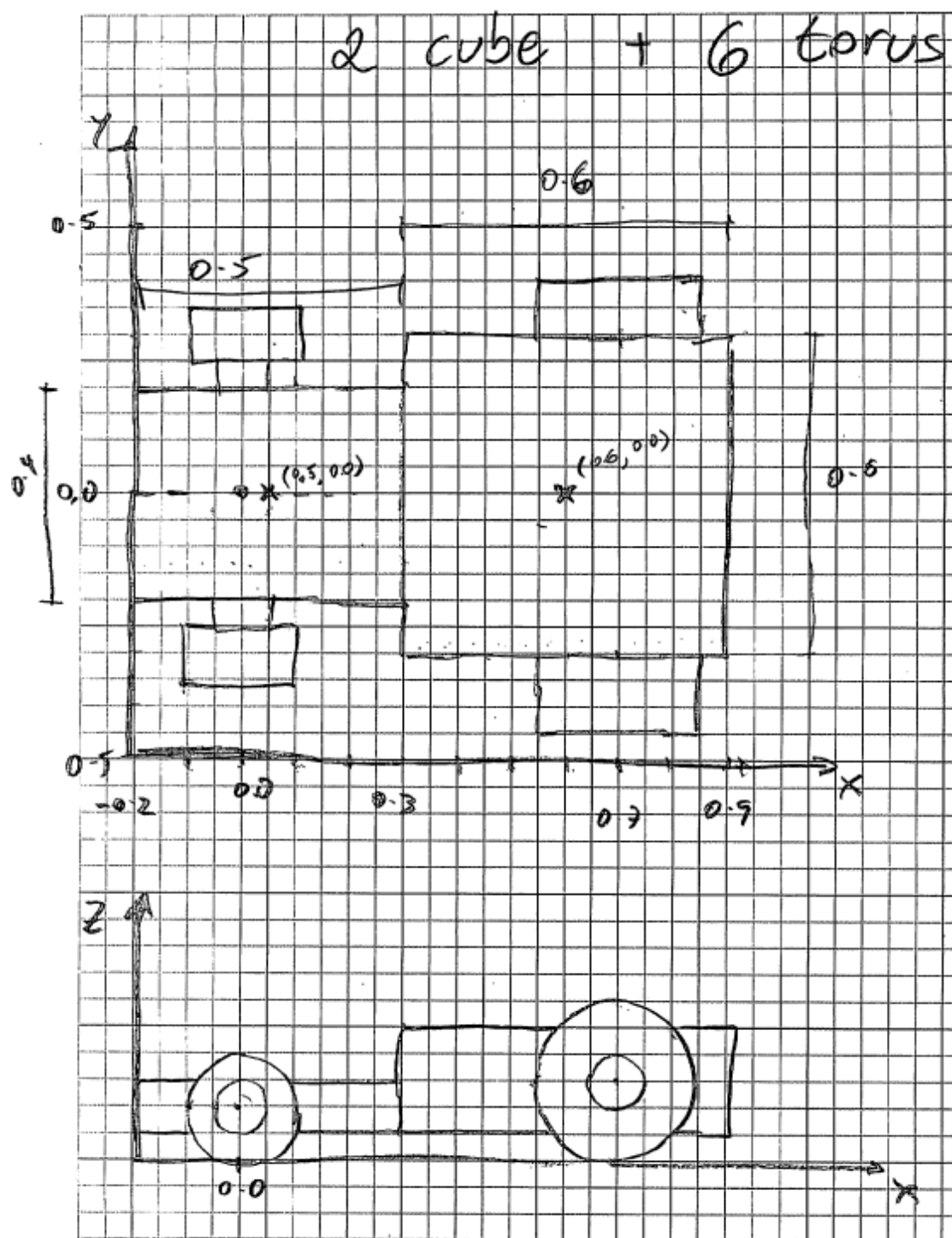


A4. Fonte de Luz



Anexo B

Esboço de Objecto



Importante: Não usar este esboço exemplificativo como base dos modelos a desenvolver no trabalho de laboratório. Os objectos apresentados devem ser desenhados pelos alunos.