

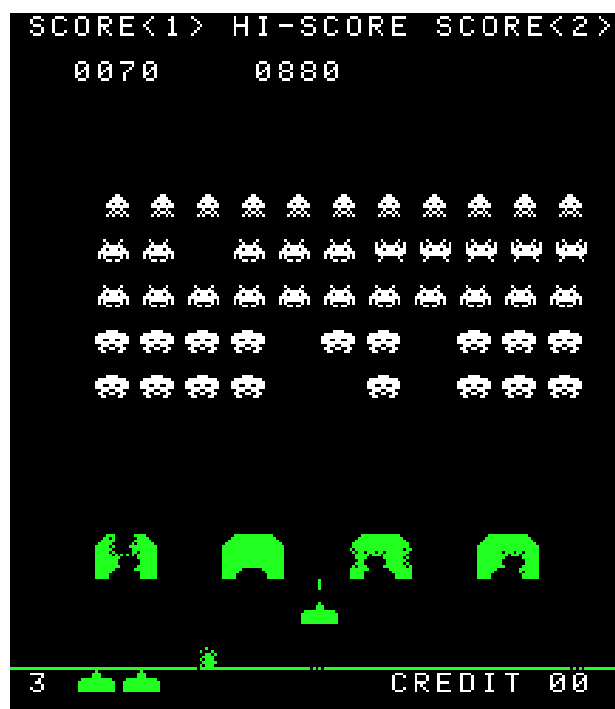
## Parte 1

### Cena Simples Interativa com Câmara Fixa

#### Breve Introdução ao Tema

O jogo Space Invaders foi criado por Tomohiro Nishikado para a Taito e foi lançado em 1978<sup>1</sup>. O jogador controla uma nave que tenta destruir ondas de alienígenas com o objetivo de ganhar o maior número de pontos possível.

Nos últimos 35 anos têm aparecido inúmeras versões deste jogo, incluindo algumas reimplementações com o grafismo original que podem ser jogadas online<sup>2</sup>.



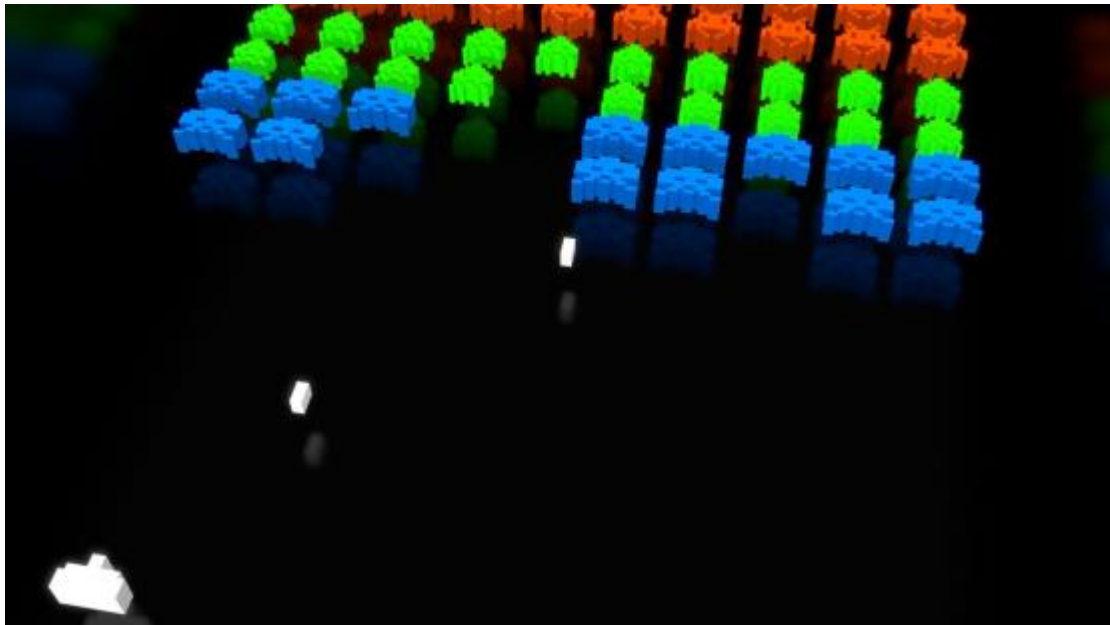
---

<sup>1</sup> [http://en.wikipedia.org/wiki/Space\\_Invaders](http://en.wikipedia.org/wiki/Space_Invaders)

<sup>2</sup> <http://www.freeinvaders.org/welcome.php>

## Trabalho a realizar

O objectivo dos trabalhos de laboratório de Computação Gráfica deste ano é recriar este clássico numa versão 3D simplificada, recorrendo a **three.js**. A ideia é manter a jogabilidade original alterando a perspectiva gráfica para que os vários elementos do jogo tenham um aspecto tridimensional. 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 **10 a 14 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**.

Não esquecer de preencher e submeter a *timesheet* correspondente com as horas despendidas pelo grupo na realização deste trabalho.

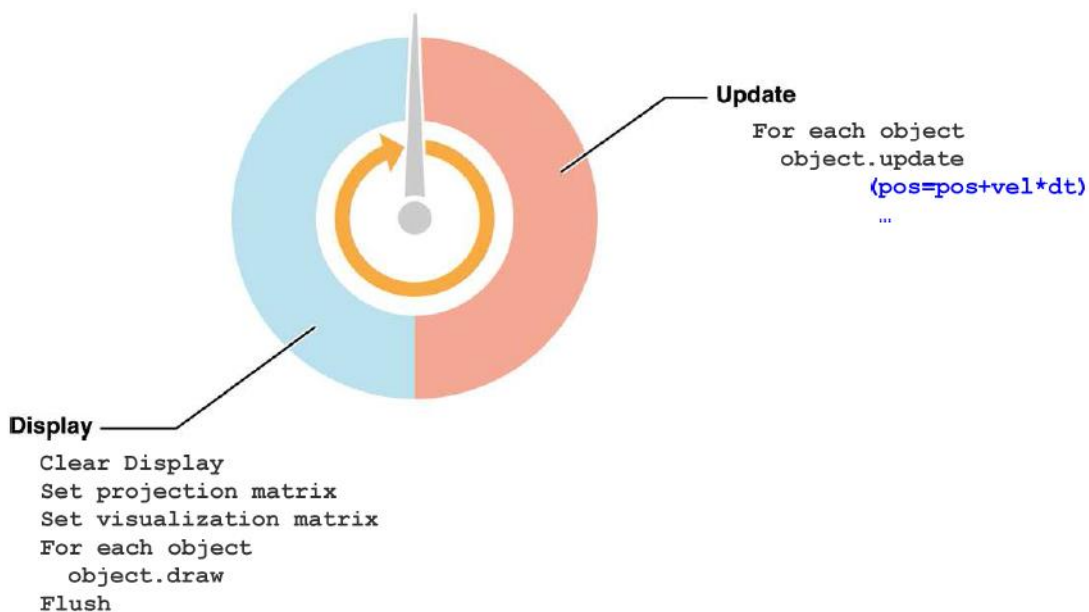
## Tarefas

As tarefas para a primeira parte são:

1. Definir uma câmara fixa com uma vista de topo sobre a cena utilizando uma projecção ortogonal (semelhante à vista 2D do jogo original). Neste primeiro laboratório, todo o campo de jogo deve ficar visível. [**1,5 valores**]
2. Modelar a nave e os alienígenas. Bastam duas fileiras de quatro, podendo ser todos de uma única espécie. Para o efeito deve-se recorrer a objetos geométricos tridimensionais simples (cubos, esferas, cilindros, cones, etc...). Cada elemento deve ser composto por três ou mais destes objectos. Os elementos devem ser desenhados em papel antes de escrever código. Estes esboços devem ser apresentados durante a avaliação. A representação de todos os elementos do jogo deve alternar entre modelo de arames e sólida usando a tecla 'A'. [**2,5 valores**]
3. Controlar o movimento da nave com o teclado utilizando as teclas das setas esquerda '←' e direita '→'. A nave não deve atingir a velocidade máxima imediatamente após pressionar a tecla ou parar quando se larga. Deve ter um movimento uniformemente variado, considerando aceleração constante. Para esta avaliação não existem colisões entre os elementos do jogo nem os alienígenas se movem. Estes serão objectivos de uma futura entrega. [**2,0 valores**]

## Informação Importante

A implementação do trabalho desenvolvido nos laboratórios de computação gráfica deve usar o ciclo de animação (update/display cycle). Este padrão de desenho, usado nas aplicações de computação gráfica interactiva, está ilustrado na figura abaixo e separa o desenho da cena no ecrã da actualização do estado do jogo em duas fases distintas. Na fase de display são cumpridos três passos base: limpar o buffer; desenhar a cena e forçar o processamento dos comandos. Na fase de update todos os objectos do jogo são actualizados de acordo com a física inerente. Para este trabalho pode-se aplicar apenas a equação do movimento linear uniforme. É ainda nesta fase que se processa a detecção de colisões e implementação dos respectivos comportamentos.



Para além de dos acontecimentos de update e display existem mais um conjunto de acontecimentos, tais como teclas pressionadas ou soltas, temporizadores e redimensionamento,. Estes devem ser tratados pelas respectivas funções de callback de forma independente.

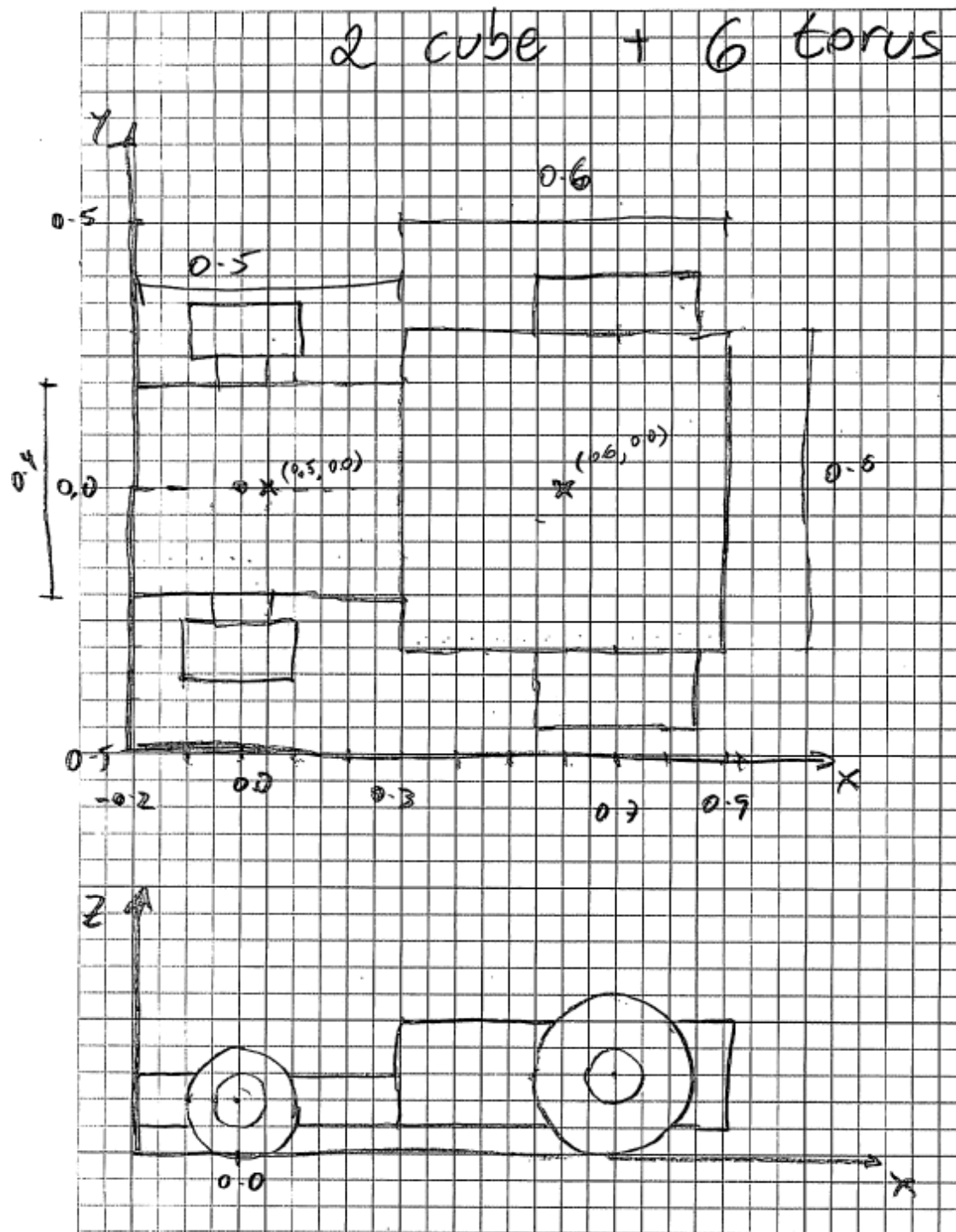
Por fim, sublinhamos que deve ser aplicada uma programação orientada a objectos e seguidas boas práticas de programação que permitam a reutilização do código em entregas posteriores e facilitem a escalabilidade futura.

## Sugestões

1. Na definição da câmara ortogonal devem desenhar um esboço em papel e não tentar definir os parâmetros “de cabeça” ou por “tentativa e erro”.
2. Para validar o posicionamento e orientação da câmara antes de implementarem os restantes elementos jogo podem usar uma primitiva (e.g. esfera) e verificar se obtém a visualização esperada.
3. Devem desenhar os objectos em papel antes de escrever código ajuda muito a perceber que primitivas e transformações devem ser aplicadas. No Anexo A pode encontrar **um exemplo** de um esboço em papel para um objecto. Devem apresentar um esboço em papel dos objectos do jogo no âmbito do segundo objectivo.
4. Antes de implementar os elementos de jogo de imediato a sua forma final, sugerimos que os implementem e visualizem primeiro como objectos simples. Por exemplo: a nave pode ser um cubo, os alienígenas cones e os escudos cilindros.

## Anexo A

### 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. Ademais, isto é um carro!