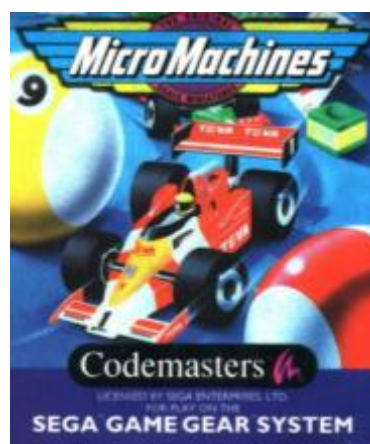


Parte 1

Cena Simples Interativa com Câmara Fixa

Breve Introdução ao Tema

O jogo *Micro Machines* foi o primeiro de uma série de jogos baseados nos famosos carros de brinquedo com o mesmo nome. Este jogo foi lançado originalmente em 1991 tendo, até à última versão em 2006, sido lançadas versões para um vasto conjunto de plataformas. O jogador controla um carro de brincar em ambientes pouco convencionais, tal como uma cozinha ou um jardim. O carro tem de percorrer uma pista, evitando os mais diversos obstáculos. Ao embater ou passar por cima de obstáculos o carro perde velocidade ou torna-se mais difícil de controlar. É possível assistir a cerca de sete minutos de gameplay¹ no canal NESguide do Youtube.



Nos últimos anos têm aparecido inúmeras versões deste jogo, incluindo algumas recodificações com o grafismo original, ilustrado

¹ <https://www.youtube.com/watch?v=BMpZznee74I>

na figura anterior, juntamente com a capa do jogo. Algumas destas versões podem ser jogadas online².

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 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

² <http://www.classicgamesarcade.com/game/21618/micro-machines-driving-racing-game.html>

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 codificar 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 **9 a 13 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 se esqueçam de preencher e submeter a *timesheet* correspondente com as horas despendidas pelo grupo na realização deste trabalho.

Tarefas

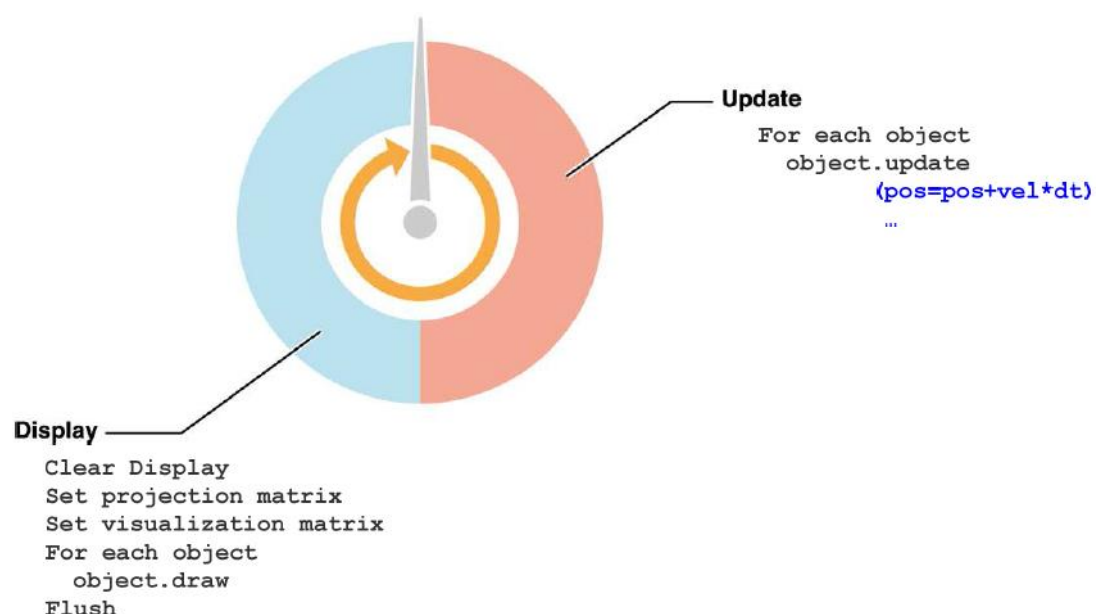
As tarefas para a primeira parte são:

1. **Modelar a mesa usando um cubo e modelar a estrada definindo as suas bermas usando pequenos *torus* (toros, em Português), simulando *cheerios*. 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, toda a pista deve ficar visível. A relação de aspecto deve ser mantida e a pista deve continuar visível quando a janela é redimensionada [1,5 valores]**
2. **Modelar o carro, três laranjas e cinco pacotes de manteiga, recorrendo a objectos geométricos tridimensionais simples (cubos, esferas e *torus*). O carro deve ser composto por mais do que um objecto geométrico, num mínimo de cinco objectos. A representação destes objectos deve alternar entre modelo de arames e sólida usando a tecla 'A'. Devem apresentar o esboço em papel dos objectos modelados [2,5 valores]**

3. Controlar o movimento do carro com o teclado utilizando as teclas das setas para virar para esquerda '←' e direita '→', assim com o para acelerar para a frente '↑' ou para trás '↓'. O carro não deve atingir a velocidade máxima imediatamente após pressionar a tecla ou parar quando se larga. O carro deve ter um movimento uniformemente variado, considerando a velocidade e a aceleração escalares, sendo a direcção do movimento dada por um vector tridimensional. [2,0 valores]

Informação Importante

A realização do trabalho desenvolvido nos laboratórios de computação gráfica deve usar o ciclo de animação (update/display cycle) conforme descrito nas aulas. 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. Estes passos são concretizados pelo método `render()` do `WebGLRenderer`. Na fase de *update* todos os objectos do jogo são actualizados de acordo com as leis do movimento inerentes a cada um. Para este trabalho pode-se aplicar apenas a equação do movimento linear uniformemente acelerado. É ainda nesta fase que se processa a detecção de colisões (numa entrega posterior) e codificação dos respectivos comportamentos.



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

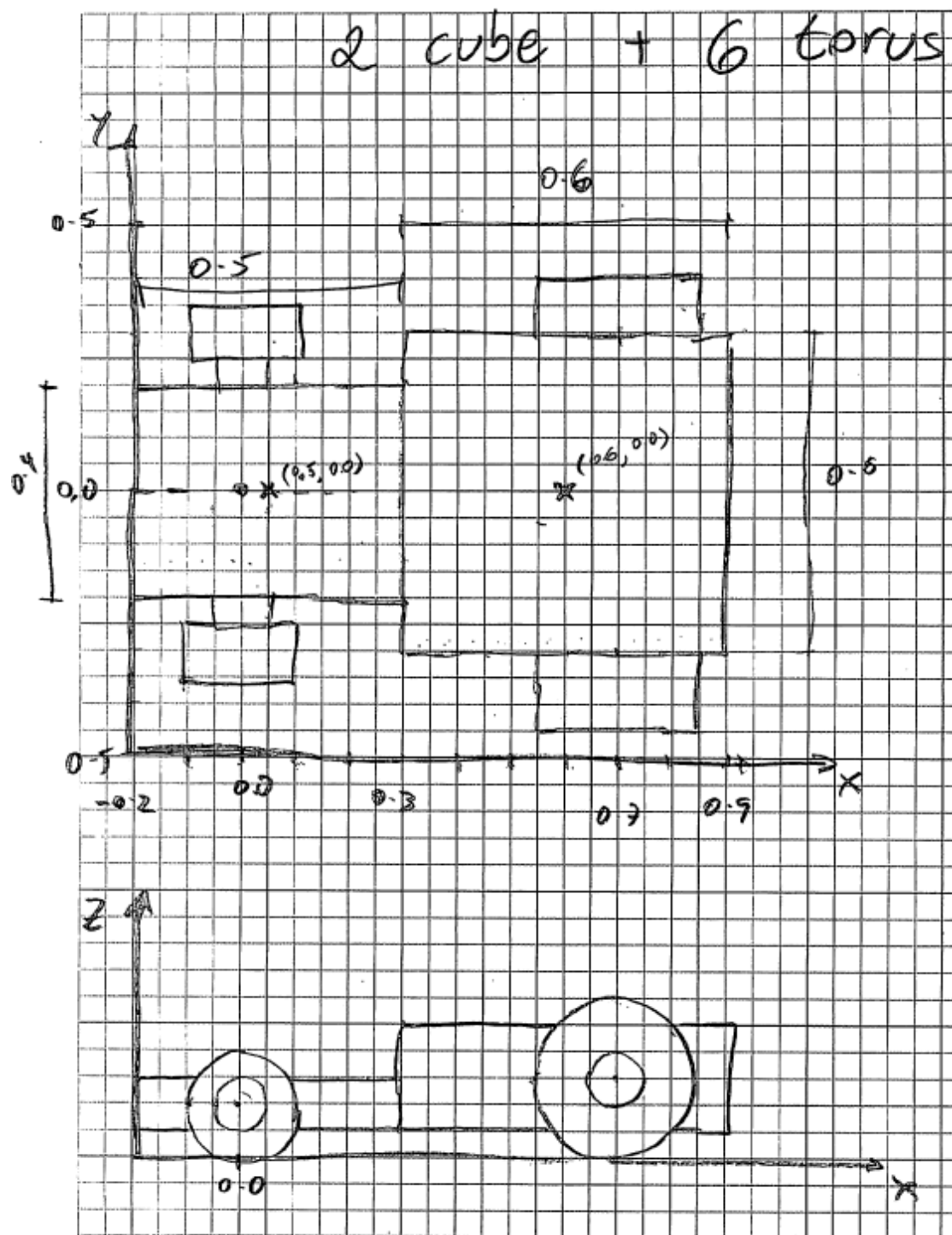
Por fim, sublinhamos que **deve ser aplicada** uma programação orientada a objectos e seguidas boas práticas de codificação que permitam a reutilização do código em entregas posteriores e facilitem a escalabilidade futura.

Sugestões

1. Desenhar os objectos em papel antes de escrever o código **three.js** 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 (**não usar** este esboço).
2. As rodas do carro podem ser criadas recorrendo a toros.
3. Para esta avaliação não existem colisões entre os objectos nem as laranjas se movem. Estes serão objectivos de uma futura entrega.
4. Antes de codificar os elementos de jogo na sua forma final, sugerimos que os modelem e visualizem primeiro como objectos simples.

Anexo A

Esboço de Objecto Carro



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