



Primeiro Trabalho

Motivação:

Realidade Aumentada (RA) é uma modalidade de visualização onde objetos virtuais são misturados ou combinados a cenas reais [1]. Essa combinação visa sobretudo acrescentar informações dentro do contexto do ambiente real onde o usuário está imerso.

Um dos grandes problemas da RA é o alinhamento dos objetos virtuais dentro da imagem da cena real. Para isso, informações sobre o a câmera utilizada na aquisição da imagem bem como a localização do objeto a ser adicionado na cena devem ser conhecidas. A ideia é que uma vez conhecida a câmera e a localização do objeto podemos, utilizando técnicas de CG, sintetizar uma imagem do objeto compatível com a cena real e sobrepor essa imagem na cena de modo realista.

A localização do objeto pode ser definida a partir de um marcador, ou seja, um padrão reconhecível dentro da cena pelo qual a aplicação de AR vai buscar. A partir da posição do marcador é possível então colocar o objeto real relativo a ele (Figura 1a). É possível também utilizar referências dentro da própria imagem para localizar onde os objetos virtuais serão adicionados (Figura 1b).

Várias bibliotecas promovem os cálculos necessários para a identificação e o alinhamento dos objetos de referencia da cena real, como o **ARToolkit** [3] e **ArUco** [4]. Com o uso dessas bibliotecas o trabalho de criar uma aplicação em RA passa a ser a renderização dos modelos virtuais que irão compor a cena real.

A visualização da cena real aumentada pode ser feita de diversas formas: através de um monitor (Figura 1a) ou tela de um dispositivo móvel (Figura 1b) ou mesmo utilizando dispositivos específicos, como o *Google Glass* (Figura 1c).



(a)



(b)



(c)

Figura 1: Exemplos de aplicações de Realidade Aumentada: (a) com marcadores; (b) sem marcadores; (c) com dispositivos especiais.

Objetivo do trabalho:

Construir um protótipo de uma aplicação de Realidade Aumentada em ambiente WEB, capaz de renderizar uma cena a partir do posicionamento de um ou mais marcadores. Sua aplicação deve ser capaz de:

1. Carregar os objetos na cena posicionando-os de forma correta em relação aos seus marcadores; (valor 2,5)
2. Os objetos devem se movimentar na cena, de forma organizada e relativos uns aos outros. Esse movimento deve ser preservado mesmo que o marcador associado a cena/objeto mude de posição; (valor 2,5)
3. A cena virtual deve conter pelo menos duas fontes de luz diferentes (valor 1,0);
4. Os objetos devem ser renderizados pelo algoritmo de *Phong* (valor 1,0);
5. Pelo menos um dos objetos da cena deve ter material heterogêneo (valor 1,5);
6. Pelo menos um dos objetos da cena deve ter uma superfície rugosa (valor 1,5);

A Implementação:

A implementação deve ser desenvolvida em linguagem Javascript/WebGL/HTML5 [7] [8], a partir do código fornecido pelo professor. Para o tratamento das funções relacionadas a RA a biblioteca **js-aruco** [5] deve ser utilizada. Outras bibliotecas só poderão ser utilizadas com o aval do professor.

Os trabalhos deverão ser desenvolvidos individualmente ou em duplas. Os códigos gerados deve ser comentados e legíveis. Acompanhando os códigos fonte um breve relatório técnico (**em formato pdf** com 2 ou 3 paginas) deve ser entregue, descrevendo brevemente o que foi implementado e como a aplicação deve ser utilizada.

Caso necessário, a interface para o controle da aplicação deve ser feita em HTML5/ Javascript.

A Entrega:

O trabalho deverá ser submetido via *Moodle*, respeitando a data e hora limite para entrega. **Em caso de atraso, será aplicado um fator de penalização de 1,0 ponto por dia de atraso.** Qualquer problema de arquivos corrompidos ou similar o trabalho será considerado não entregue. Portanto, verifique bem o que for entregar.

Os arquivos devem ser enviados seguindo o seguinte padrão: arquivo compactado (zip, rar, tgz ou gzip apenas) contendo um diretório com o nome do(s) aluno(s) e seu arquivos. Arquivos fora desse padrão sofrerão penalização de 0,5 pontos na nota final.

Códigos com caminhos absolutos ou qualquer outra pendência que impeça a execução não serão avaliados.

A cooperação entre alunos e grupos é considerada salutar. No entanto, trabalhos com alto grau de similaridade serão tratados como “plágio”, o que resultará em avaliação **zero para todos os envolvidos**.

Qualquer dúvida adicional, evite problemas: não presuma nada, procure o professor para esclarecimentos.

Referencias Bibliográficas:

- [1] Azuma, Ronald. **A Survey of Augmented Reality Presence: Teleoperators and Virtual Environments**, pp. 355–385, August 1997.
- [3] **ARToolkit - Optical marker tracking and overlay for augmented reality;**
<http://sourceforge.net/projects/artoolkit/>.
- [4] **ArUco: a minimal library for Augmented Reality applications based on OpenCv;**
<http://www.uco.es/investiga/grupos/ava/node/26>.
- [5] **js-aruco - JavaScript library for Augmented Reality applications;**
<https://code.google.com/p/js-aruco/>.
- [6] Hughes , J.F., van Dam, A., McGuire, M., Sklar, D.F., Foley, J.D., Feiner, S.K., Akeley, K., **Computer Graphics: Principles and Practice**, Addison-Wesley Professional, 3rd Edition, 2013.
- [7] Matsuda,K., Lea, R., **WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL**, Addison-Wesley Professional, 1st edition, 2013.
- [8] Parisi, T., **WebGL: Up and Running**, O'Reilly Media, 1st edition, 2012.