

Universidade Federal de Santa Maria  
Curso de Ciência da Computação  
Disciplina: Computação Gráfica Avançada  
Segundo Semestre de 2016  
Prof. Cesar Tadeu Pozzer  
Data: 08/11/2016

## Trabalho 4 – Unity

Desenvolva um programa com a engine Unity que use elementos 3D e elementos de interface (Canvas). O programa deve implementar a movimentação de um personagem andando em um terreno 3D, algum tipo de simulação física (e colisão), como disparo de objetos contra elementos móveis.

Deve também dispor de elementos de UI, incluindo: botão, slider e scrollbar. Cada evento feito no “jogo” deve gerar uma mensagem (log), que deve ser exibida no scrollbar. Cada mensagem deve ser adicionada ao log como um prefab. Deve-se ajustar a barra de rolagem para refletir a quantidade de elementos que estão presentes no scrollbar.

Procure explorar o uso de prefabs, gameobjects, scripts, etc.

O programa não precisa ter recursos de IA.

### Data e Formato de Entrega

- Data: 15/nov/2016, 23:59h.
- No email e no cabeçalho do arquivo devem conter o nome completo do aluno.
- O arquivo deve ser enviado para [pozzer3@gmail.com](mailto:pozzer3@gmail.com), [mdalcin@inf.ufsm.br](mailto:mdalcin@inf.ufsm.br) com o subject “CGA T4”.
- Envie **apenas** os arquivos de projeto Unity, código fonte e modelos/texturas.
- O programa deve ser enviado em um arquivo compactado fulano.rar (fulano = login ou nome do aluno). Dentro deste arquivo deve haver um diretório com o mesmo nome do arquivo e dentro deste diretório os arquivos do trabalho.
- Ex: o arquivo pozzer.rar deve conter um diretório chamado pozzer, e dentro do diretório devem estar os arquivos do trabalho.

### Critério de Avaliação

- Documentação: descrever no cabeçalho de cada arquivo a ideia geral do código e detalhes específicos de partes que mereçam uma explicação – não comente por exemplo o que faz b++.

- README.txt: incluir um arquivo "README.txt" contendo informações sobre quais funcionalidades foram implementadas (requisitos e extras).
- Pontualidade: Trabalhos não entregues na data não serão avaliados e receberão nota zero.
- Legibilidade: nome de variáveis, estruturação do código. O código digital a ser entregue deve ter 4 espaços de indentação e não deve possuir tabulações.
- Clareza: facilidade de compreensão – evite códigos complexos e desnecessários. Adote a solução mais simples possível.
- Funcionalidade: o programa deve satisfazer todos os requisitos. Programas que não compilarem ou que não atenderem nenhum requisito receberão nota 0 (zero).

Você pode discutir estratégias e ajudar o colega na implementação, porém evite passar código fonte. Programas semelhantes terão nota 0 (zero).