

**Trabalho Prático**  
**Fundamentos de Computação Gráfica - *INF01047***

Cleiton Souza Lima - 00262511  
Felipe De Mello Flores - 00274712

Professor Dr. Eduardo Simões Lopes Gastal

# Sumário

---

- Descrição do Jogo
- Instruções
- Conceitos aplicados
  - Criação de objetos a partir de arquivos .OBJ
  - Aplicação de matrizes de transformações lineares
  - Movimentação
  - Aplicação de modelos de iluminação
  - Aplicação de Texturas
  - Intersecção de objetos
  - Câmera
- Conclusão

## Descrição do Jogo

---

“The Castle Defensor” é um jogo de tiro em primeira pessoa cujo objetivo é impedir que os “monstros cúbicos”, liderados pelas vacas gigantes, destroem o muro de proteção e invadem o castelo do reino.

Equipado com uma arma, o Defensor atinge os inimigos que saem da grama alta destruindo-os antes que cheguem até o muro.

## Instruções

---

A movimentação do personagem é realizada utilizando as teclas “w”, “a”, “s”, “d” e, com o botão esquerdo do mouse, são disparados tiros contra os inimigos. Para saltar, utilize a tecla ‘espaço’.

Os monstros possuem 3 de vida e as vacas 30. O tiro causa 1 de vida.

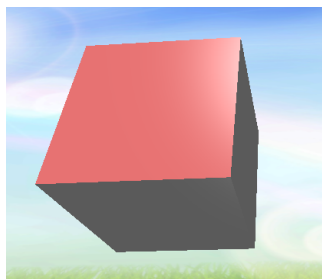
O muro possui 10 de vida, sendo que cada monstro que chega nele causa 1 de vida.

Se uma vaca chegar, ele é destruído.

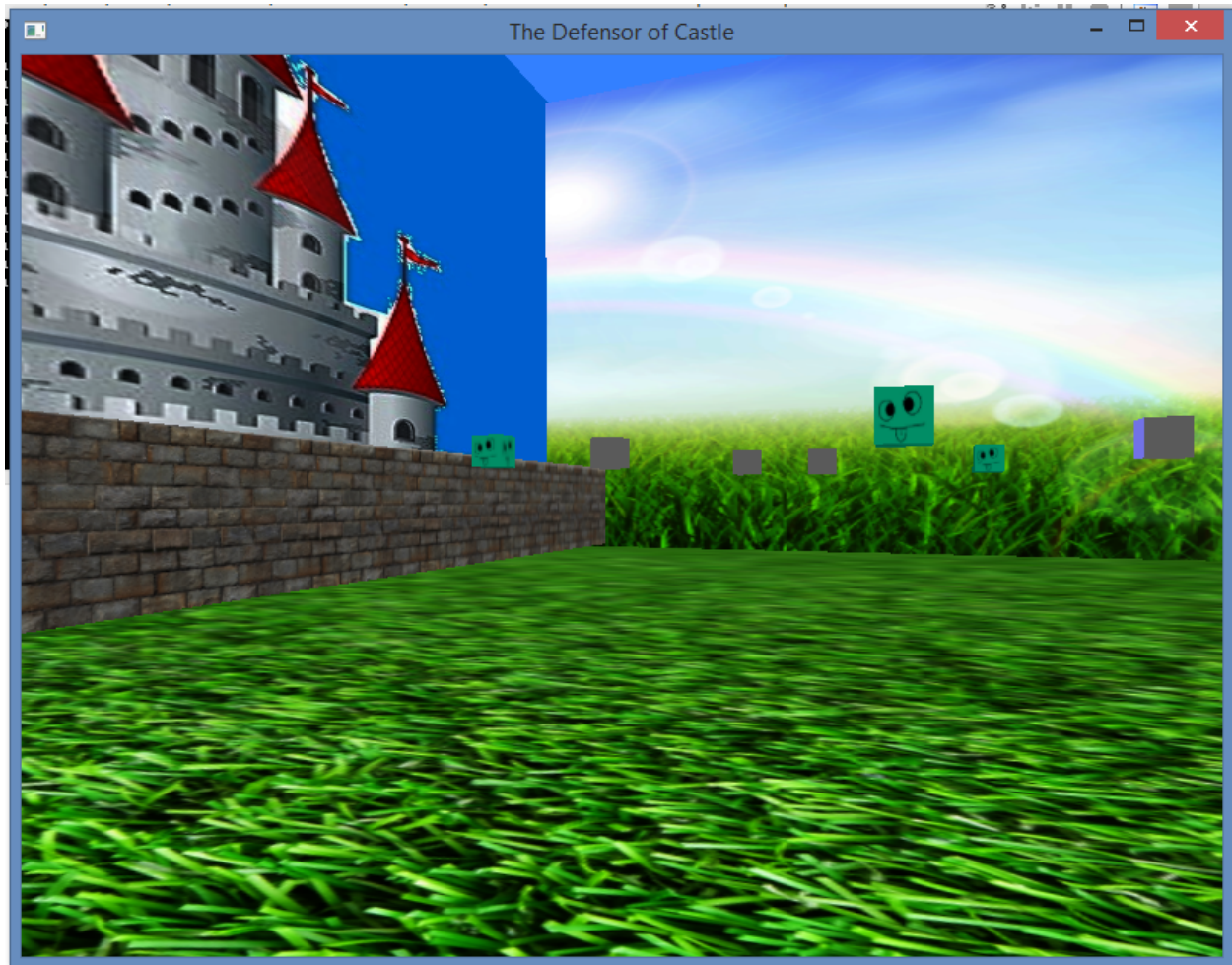
## Conceitos Aplicados

---

- Com o intuito de manter o código organizado, todos os objetos utilizados nas cenas foram importados a partir de arquivos OBJ. No caso de objetos com alguma textura aplicada, suas coordenadas também foram armazenadas no arquivo antecedido do identificador “vt”. Para posicioná-los na cena, as seguintes matrizes de transformação foram aplicadas: Escalamento, Translação e Rotação. Além disso, para a posição dos cubos ser definida de forma aleatória, foi gerador valores com a função “SRand” para a coordenada X (no plano cartesiano) do objeto.
- Todos os movimentos são baseados no tempo, fazendo com eles fiquem constantes independente do hardware.
- Quanto a iluminação, os cubos azuis são iluminados de forma difusa utilizando a lei dos cossenos de Lambert e calculado uma vez para cada vértice(modelo Gourad). Os cubos vermelhos, além do termo difuso calculado por pixel(modelo Phong), é aplicado uma constante especular calcula usando o modelo de Blinn-Phong. Em ambos, uma constante de iluminação ambiente é aplicada. Por fim, os cubos verdes estão somente com uma textura aplicada, sem nenhum tipo de iluminação. Além disso, estamos fazendo o uso de uma fonte de luz pontual.



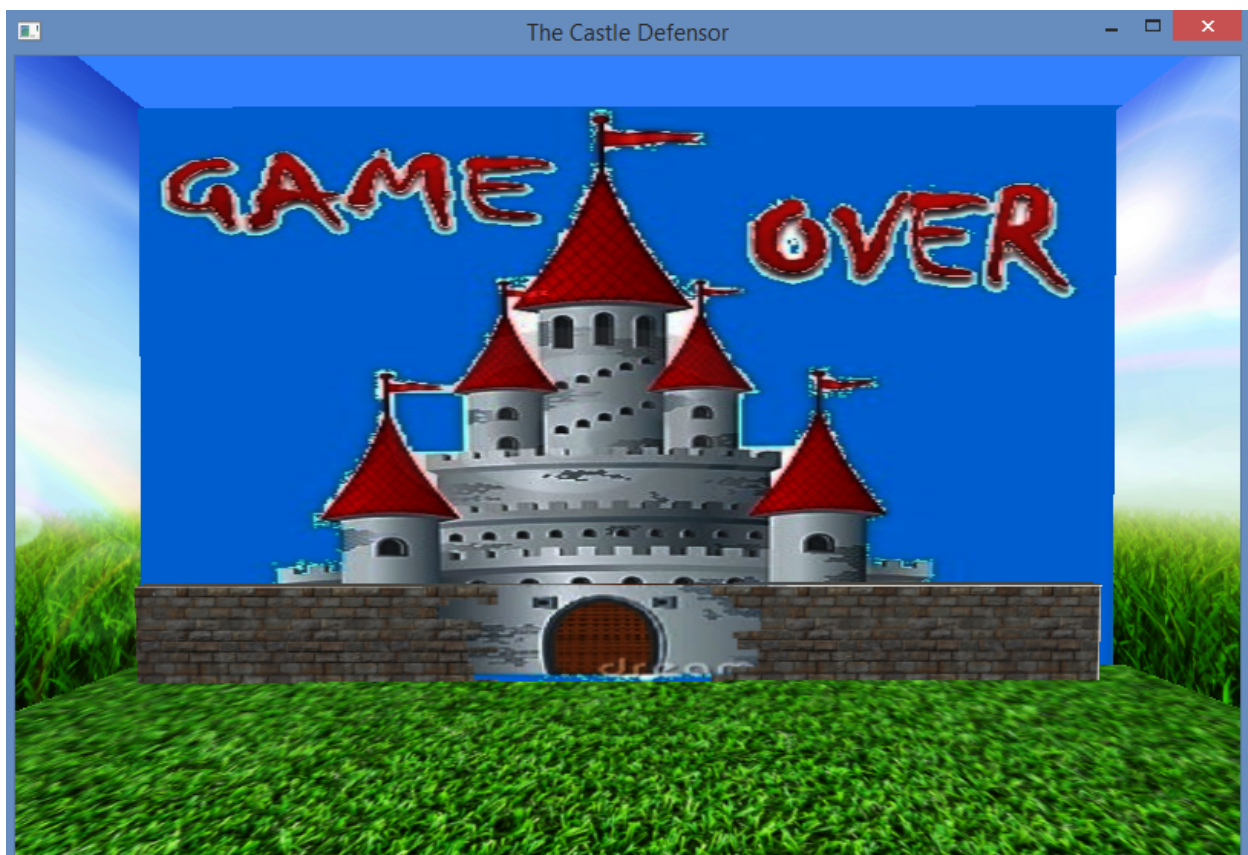
- A cena virtual foi montada utilizando planos com diferentes texturas, representando o chão e os limites da cena. Também há um plano representando o muro, objeto no qual calculamos as intersecções e monitoramos a “vida” do muro.



- Existe o teste de intersecção entre as balas com os objetos e os monstros na cena. Também há testes entre o personagem com os objetos, com os monstros e com o muro. Todos os testes são do tipo AABB-AABB, com exceção do caso do personagem que é usado o teste AABB-PONTO e PLANO-PONTO. No caso da balas, apesar de elas serem redondas, seu tamanho são bem pequeno, não fazendo muita diferença o tipo de teste. As vacas apesar de serem um objeto complexo, são tratadas suas intersecções como sendo AABB. Foi usado um teste sem considerar a movimentação, pois os monstros são lentos e grandes e as balas não são tão rápidas e são pequenas, o que leva a nunca acontecer de acabar não

detectando a interseção por causa do movimento mesmo em máquinas mais lentas. Foi testado em máquinas do laboratório de informática e outros computadores lentos.

- Em relação ao movimento do personagem, há dois tipos de teste de interseção, o horizontal e o vertical. O Vertical para os casos de pulo ou para simular gravidade e o horizontal para permitir que o personagem não entre em objetos. É usado o teste AABB-PONTO para testar com os objetos e PLANO-PONTO para testar com o chão.
- Nossa aplicação utiliza dois tipos de câmeras: Free Camera durante a partida, possibilitando ao usuário melhor movimentação pela tela e, após a queda do muro(Game Over), a câmera é alterada para uma Look-At fixa no castelo, como pode-se ver na imagem abaixo:



## Conclusão

Acreditamos ter atingido o objetivo do trabalho prático: utilizamos diversos conceitos vistos em aula e, embora já houvésssemos tido contato com a maioria deles nos laboratórios, o grande desafio foi unir todos em uma aplicação coesa e com sentido lógico. A parte de intersecção de objetos foi o grande desafio, uma vez que, para cada nova forma que pretendíamos utilizar na cena, um novo algoritmo para detectar intersecções seria necessário. Assim, limitamos nossa aplicação a intersecções do tipo AABB-PONTO para a movimentação do personagem e AABB-AABB para o resto. Foi analisado que mesmo o teste não sendo perfeito são casos específicos que ele falha e atinge muito satisfatoriamente o contexto da aplicação, no caso o jogo.