

# Infernal Rampage

---

Éder Monteiro e Tiago Binz

# Doom-like



# Interação por mouse e teclado

As interações por mouse e teclado são feitas através de uma classe com membros estáticos chamada Input. Ao inicializar o programa, os callbacks são apontados para esta classe. Isso facilita o acesso para checagens de input em todas classes sem necessidade de variáveis globais.

# Matrizes

A aplicação implementa transformações geométricas através de um sistema de Game Objects e Transforms.

A câmera é definida através de uma subclasse de Game Object, o PlayerController, que possui informações como Near Plane, Far Plane e Field of View.

# Objetos virtuais

As malhas de triângulos foram, em sua maioria, baixadas do Models Resource, um site que disponibiliza modelos 3D de jogos lançados para projetos não comerciais



# Transformações geométricas de objetos virtuais

A classe `Input` fornece informações para qualquer objeto sobre a entrada atual que o usuário está dando (movimento para frente, tiro, movimento da câmera, etc.). A classe `PlayerController` chama funções de transação e rotação do objeto através dessas informações.

Enquanto o jogador segura alguma das teclas de movimento, a arma usa uma função seno para se deslocar para cima e pra baixo, dando noção de movimento com passos.

# Câmera

O jogo apresenta um controle de câmera convencional para jogos do gênero First Person Shooter. Há também uma feature especial para implementar a câmera look at. Ao segurar o botão direito do mouse, a câmera fica acompanhando um inimigo específico, o que facilita para acertá-lo.

# Múltiplas instâncias

No código fonte da aplicação existe uma diferença entre um Game Object e uma Mesh. Um Game Object desenha uma instância de uma Mesh que só precisa ser carregada uma vez pelo programa, que recebe um “nome”, ou “índice” pelo qual ela pode ser referenciada.





# Intersecção entre objetos virtuais

- Cápsula-esfera: implementa a colisão entre um personagem (jogador ou inimigo) e um projétil.
- Cápsula-plano: implementa a colisão entre o chão e o jogador
- Plano-esfera: implementa a colisão entre um projétil e o chão do cenário

# Modelos de iluminação

- Lambert: todos objetos do jogo, exceto a arma
- Blinn Phong: somente a arma do jogador
- Gourad (interpolação): todos objetos do jogo, exceto arma e power up
- Phong (interpolação): arma do jogador e power up

# Mapeamento de texturas

O mapeamento das texturas foi feito no programa Blender.

Os objetos trazidos de outros jogos já tinham parte dessa informação pronta.

Nós decidimos que o jogo ficaria mais dinâmico se os mapeamentos UV pudessem ser alterados em tempo de execução, para simular movimento da lava, contar o número de balas do jogador como no jogo Halo e apresentar efeitos de explosões animadas.

# Curvas de Bézier

Curvas de Bézier foram utilizadas para implementar a movimentação dos inimigos. Cada objeto representando um inimigo possui uma movimentação independente dos demais, baseada na sua curva de Bézier específica.

# Animação baseada em tempo

Utilizamos os conceitos vistos em aula para implementar as animações do jogo. No loop principal da classe `World`, todos objetos são renderizados, e realizam uma chamada da função `Update`, recebendo como parâmetro o `Delta Time`, ou seja, a quantidade de tempo em segundos que se passou desde o último quadro.

# Efeitos sonoros

A implementação de efeitos sonoros foi feita utilizando a biblioteca irrKlang.

# Como jogar

```
DEMON INVASION!  
Earth has been invaded once again by Satan's bloodthirsty demon legions.  
This Archdemon's destructive power has been measured as 1050  
You must save Mankind!
```

```
HOW TO PLAY  
Use the mouse to look around.  
Press the left mouse button to fire.  
Use the W A S D keys to move.  
Hold the left mouse button to lock on to a target.
```

```
PRESS ENTER TO START THE FIGHT...
```

# Como compilar

Tudo que é necessário para compilar a aplicação e jogar está dentro do .zip fornecido. O projeto de Code Blocks já fez o “link” adicional à biblioteca irrKlang para que os efeitos sonoros funcionem. Basta abrir o projeto nessa IDE e compilar.