

**TRABALHO PRÁTICO DA DISCIPLINA DE CG & RV – 2025/2**  
**BATALHA ESPACIAL**

**Descrição:**

O **trabalho** consiste em desenvolver um **jogo 2D ou 3D simples** de batalha espacial, a ser desenvolvido, única e exclusivamente, **em trios**. Para a implementação, recomenda-se o uso da game engine Unity e o uso de modelos com geometria simples. A plataforma é livre (PC ou mobile).

O objetivo do jogo é destruir diferentes alvos (aviões inimigos e asteroides) através de tiros durante determinado período de tempo. O piloto (usuário) deve guiar o avião evitando a colisão com os alvos. Os aviões inimigos podem se deslocar de diversas maneiras pelo espaço, de maneira análoga ao movimento da câmera. Os asteroides mantêm uma posição fixa.

O jogo deve ter, no mínimo, dois níveis de dificuldade, que variam de acordo com o número de aviões inimigos a abater (mínimo = 5 aviões). É livre explorar outras variações de dificuldade, como velocidade das aeronaves inimigas, uso de neblina, etc. A escolha do nível deve ser feita em um menu de opções, que também dá início ao jogo e mostra um ranking simples.

Neste jogo, existe uma área limitada dentro do qual os aviões podem voar (ou seja, o avião do jogador precisa cruzar uma área, de um ponto de partida para um ponto de chegada). A área de ter, no mínimo, o tamanho de 30 aviões enfileirados, conforme o tamanho relativo utilizado para o avião.

O avião do jogador deve ter uma aceleração constante, que pode ser aumentada ou diminuída. Os movimentos de deslocamento consistem, basicamente, de translações e/ou rotações que correspondem à manipulação da câmera através do teclado (andar para frente, reduzir velocidade, girar para baixo ou para cima). Não é necessário desenhar o painel de instrumentos do avião.

Através do mouse (ou de uma tecla ou de um toque na tela), deve ser possível atirar na direção do deslocamento da câmera. Cada vez que o usuário atira, o objeto que representa o projétil deve aparecer na frente da visão do usuário e se deslocar (em linha reta) até que atinja o inimigo ou saia da área limitada. Para realizar o deslocamento do projétil, utilize o mesmo princípio de movimentação da câmera para frente.

O jogo acaba quando o número mínimo de inimigos destruídos é atingido, e o avião do jogador chega no destino, obedecendo um tempo pré-determinado para concluir a tarefa. Caso contrário, o jogador perde a batalha.

**Objetivos:**

- Consolidar o conhecimento sobre representação, manipulação e visualização espacial de objetos;
- Exercitar conceitos de Computação Gráfica, como navegação em ambientes e detecção de colisão;
- Incentivar a criatividade do aluno, através da construção de cenários 2D ou 3D;
- Incentivar o aprendizado de recursos de uma *game engine*, explorando a inclusão de *assets*, carga de objetos, detecção de colisão, efeitos e *timer*.

**Apresentação:**

Os trabalhos serão apresentados em laboratório, abertamente à toda a turma, no formato **pitch**. A solução deverá ser exibida funcionando corretamente, dentro de **5 minutos**, com domínio do trio para explicar o projeto ao professor.

**Compartilhamento:**

Obrigatoriamente, deve-se criar um repositório do projeto no GitHub, compartilhando com o professor (rafaelrieder) para acompanhar o progresso e o envolvimento da equipe. O local deve compartilhar somente o código-fonte e dependências pertinentes ao projeto. O README do projeto deve apontar *links* para baixar o executável do jogo, e para visualizar o vídeo demonstrativo da solução.

- **Todos os componentes do grupo devem executar *commits* no repositório!**
- **Hospedar o projeto no repositório git é “eliminatório” para avaliação do trabalho da disciplina.**

**Datas importantes:**

29/09, 03/11, 10/11 e 17/11: sprints (tempo em aula para desenvolvimento do projeto).

24/11/2025: apresentação final do trabalho.

**Critérios de avaliação do grupo:**

- Menu principal, com níveis de dificuldade, ranking e sobre o jogo (0,5);
- Qualidade da modelagem das aeronaves e do cenário: quantidade e variedade de objetos (1,0);
- Movimentação correta da câmera do usuário, comportamento dos inimigos, asteroides e tiros (2,5);
- Nível de interatividade: facilidade de controle, velocidade de execução, detecção de colisão (2,5);
- Utilização de efeitos visuais e recursos sonoros (1,0);
- Demo em vídeo (0,5).

**Critérios de avaliação individual:**

- Contribuições no GitHub e domínio na apresentação (2,0).

**Pontos-extras:**

- Suporte a Realidade Virtual ou Realidade Aumentada (2,0) – podem ser cumulativos para as outras avaliações da disciplina.

Casos omissos ou dúvidas podem ser consultados pessoalmente ou por e-mail: [rieder@upf.br](mailto:rieder@upf.br)