

# Run 'n Gun

## INTRODUÇÃO

Jogos do tipo run and gun são um subgênero dos jogos de ação e tiro, caracterizados por uma jogabilidade rápida, intensa e baseada em reflexos, onde o jogador corre (run) e atira (gun) ao mesmo tempo, geralmente contra grandes ondas de inimigos e obstáculos em movimento constante.

Um bom exemplo de run and gun é o jogo chamado Contra. Ele é um clássico jogo do gênero, lançado pela Konami em 1987, que se tornou um dos títulos mais icônicos dos arcades e do console NES. Você pode ver uma gameplay de Contra clicando [AQUI](#).

Outro exemplo, este bastante popular até hoje, é o jogo Metal Slug. O Metal Slug foi criado pela SNK e lançado originalmente em 1996 para os arcades da plataforma Neo Geo. Conhecido por seus gráficos em pixel art detalhados, animações fluidas e humor característico, a série coloca os jogadores no papel de soldados que enfrentam exércitos inimigos, veículos de guerra e ameaças alienígenas. Uma gameplay de Metal Slug está disponível [AQUI](#).

## O TRABALHO

Esse será um desenvolvimento mais “livre” e podemos compreender o objetivo final como **DESENVOLVER UM JOGO GUN 'N RUN PARECIDO COM O CONTRA OU METAL SLUG** (em termos de jogabilidade e mecânicas - no aspecto criativo e temático, vocês podem implementar ideias diferentes).

Porém, para referência, vamos estabelecer apenas um conjunto de requisitos mínimos para o jogo, sendo que a implementação destes equivale a 80% da nota total do trabalho e funcionalidades extras só serão avaliadas dada a implementação dos requisitos mínimos.

O restante da nota pode ser completada através da implementação de funcionalidades extras. Existe um conjunto de sugestões de funcionalidades extras fornecidas no contexto deste enunciado; porém, seja criativo, e sugira novas funcionalidades ao seu professor (para definir a quantidade de pontos que esta valerá no contexto do trabalho).

A nota do trabalho será, no máximo, 100 pontos (que valem 30 pontos na média da disciplina). Porém, você pode implementar funcionalidades extras que somem até 110 pontos. Ou seja, se você tiver algum erro em alguma funcionalidade que reduza sua nota, ainda é possível alcançar a nota 100.

O programa deve ser bem modularizado, com uma estrutura lógica de diretórios, com uma separação organizada de funções e arquivos.

## REQUISITOS MÍNIMOS

Em relação às **características gerais** do jogo, deve ser considerado o seguinte:

- Implementação de um jogo singleplayer;
- Implementação de um menu inicial, permitindo iniciar ou sair do jogo;
- Implementação de uma tela de fim de jogo, indicando se o final foi alcançado com sucesso ou não (game over).

Em relação ao **personagem principal**, deve existir, pelo menos:

- Sprite 2D (imagem) em pé;
- Sprite 2D (imagem) abaixado;
- Sprite 2D (imagem) pulando;
- Sprite 2D (imagem) andando em pé;
- Sprite 2D (imagem) atirando;
- Mobilidade geral (andar e atirar para todos os lados, assim como pulando);
- Ataque com uma arma (de projétil - o projétil deve ter uma sprite).

Em relação à **batalha**, espera-se, no mínimo, as seguintes funcionalidades:

- Sistema de pontos de vida que reduz progressivamente conforme os golpes recebidos por um personagem.

Em relação aos **cenários**, espera-se, no mínimo:

- Um cenário com background de imagem;
- Rolling background (o cenário muda conforme o jogador se movimenta).

Em relação aos **inimigos**, espera-se, no mínimo:

- Pelo menos um tipo de inimigo normal, que lança projéteis, com suas respectivas sprites, ataque e movimentação;
- Pelo menos, um inimigo chefe, diferente do inimigo normal, com suas respectivas sprites, ataque e movimentação.

Em relação às **fases**, espera-se, no mínimo:

- Pelo menos uma fase;
- Pelo menos 6 inimigos espalhados pela fase;
- Pelo menos 1 chefe que deve aparecer após a derrota dos 6 inimigos.

## **FUNCIONALIDADES EXTRAS**

[1 - 10 pontos] Implementar, pelo menos, um ataque especial, com animações (sprites) dedicadas para o personagem principal.

[2 - 15 pontos] Implementar, pelo menos, um inimigo normal extra com animações (sprites) e ataque diferente daquele do programa base.

[3 - 5 pontos] Implementar um botão de pausa para a batalha.

[4 - 10 pontos] Implementar a função de “abaixar” e “atirar abaixado”.

[5 - 20 pontos] Implementar uma segunda fase, com background diferente e, pelo menos, um chefe diferente e inédito.

[6 - 20 pontos] Implementar um mecanismo de salvar e carregar o jogo.

[7 - 10 pontos] Implementar um sistema de estamina (o jogador “cansa” ao atacar).

[8 - 10 pontos] Remapeamento livre de teclas de ação no menu de opções.

[9 - 5 pontos] Dificuldade ajustável no menu de opções (Fácil, Médio e Díficil).

[10 - 7 pontos] Trilha sonora na tela principal, seleção de fase e na fase, com ajustador de volume no menu de opções.

[11 - 10 pontos] Tela de seleção de fase análoga a do Megaman (ou seja, implementar sistema de fases não lineares e menu de seleção para escolher as mesmas - pelo menos uma ramificação).

[12 - 5 pontos] Item coletável de "coração" que aumenta a barra de vida.

[13 - 5 pontos] Item coletável de bota com pena que desbloqueia a habilidade do jogador fazer pulo duplo.

## REQUISITOS TÉCNICOS E ENTREGA

Considere as seguintes informações para executar e entregar o trabalho:

- Você deve utilizar a linguagem C para a implementação do trabalho;

- E a biblioteca Allegro para a implementação da interface gráfica;

- Você deve preparar um *makefile* (pelo menos, com operações *make* e *make clean*) para a compilação do seu programa;

- Você deve observar todos os requisitos de entrega do trabalho elencados no programa da disciplina, não cumprir os mesmos acarretará um desconto de 5/100 na sua nota final;

- Não esqueça de tratar possíveis erros.