

6. Power ups

Os power ups são elementos que aparecem durante a execução do jogo e, ao serem coletados pela nave do jogador, aplicam efeitos temporários que melhoram seu desempenho, como maior frequência de disparo ou invulnerabilidade a danos.

Na implementação original do projeto, não havia um sistema de power-ups integrado. Por isso, foi criada no pacote classes uma classe abstrata chamada `PowerUp.java`, que encapsula atributos e comportamentos comuns entre os diferentes tipos de power-ups, como coordenadas (x, y), velocidade, raio e a verificação de colisão com o jogador. Essa classe também define a interface comum com os métodos `desenha`, `atualizaEstado` e `aplicarEfeito`.

A partir disso, foram criadas dois `PowerUps`:

- `PowerUpTiroRapido.java`: Triângulo amarelo que ao ser coletado, reduz o intervalo de tempo entre os disparos do jogador (duração de 5 segundos).
- `PowerUpInvulnerabilidade.java`: Triângulo laranja que ao ser coletado, torna o jogador temporariamente invulnerável a danos (duração de 5 segundos).

Cada classe implementa seu próprio método de desenho, definindo a cor e formato visual (triângulo com cores distintas), e personaliza o comportamento no método `aplicarEfeito`.

Além disso, a lógica de aparecimento dos power-ups foi incorporada ao sistema de leitura dos arquivos de configuração das fases. Dessa forma, o aparecimento dos power-ups é controlado por arquivos `.txt`, seguindo a mesma estrutura usada para definir o surgimento de inimigos e chefes.

Por fim, a manipulação dos power ups foi integrada na `Main.java` com uma lista do tipo `LinkedList<PowerUp>`, onde cada instância é atualizada, desenhada e removida se sair da tela ou for coletada. Essa estrutura garante a organização e reaproveitamento de código.

8 .Fundo/Background

O fundo é basicamente composto por estrelas, sendo que na implementação anterior/original haviam dois tipo de estrelas: do primeiro plano e de segundo plano, que possuem as mesmas variáveis com a diferença de que a velocidade das estrelas do primeiro plano são mais rápidas do que a do segundo plano. Além disso, no código original, essas duas estrelas estavam declaradas usando arrays.

Então para compactar o código e organizá-lo, implementamos no pacote classes, a classe abstrata Estrela.java que define os atributos comuns às estrelas, como coordenadas (x, y) e velocidade, além dos métodos mover e desenhar.

A partir disso, extendemos a classe abstrata Estrela.java em outras duas classes que seriam EstrelasPlano1.java e EstrelasPlano2.java, onde cada uma define sua própria velocidade e personaliza o método de desenho conforme necessário.

Com isso, na Main.java substituímos os arrays por ArrayList<Estrelas> da coleção para armazenar e manipular estrelas, além disso as listas são inicializadas com a mesma quantidade de estrelas da versão anterior, sendo que agora o código é mais conciso e menos redundante, reduzindo assim a sobrecarga que a função Main havia na implementação anterior.