

# Jogo do tipo Coffeebreak Roguelike

Laboratório de Algoritmia 1/Laboratórios de Informática II

## Introdução

O projeto deste ano consiste em implementar um jogo do tipo Roguelike. Os Roguelikes são assim chamados por causa do jogo Rogue criado nos anos 80 do século passado e são conhecidos por:

- Serem baseados em turnos
- Serem difíceis
- Terem gráficos simples
- Serem bastante táticos
- Só terem uma vida

No nosso caso, vamos criar um jogo simplificado do tipo Coffeebreak Roguelike que é caracterizado por ter regras simples, ser rápido de jogar e, obviamente, ser particularmente mortífero.

## O jogo

O cenário do jogo é um tabuleiro (por exemplo de 10x10) em que cada célula se encontra num dos seguintes estados:

- Está livre;
- Contém o jogador;
- Contém um inimigo;
- Está bloqueada.

Em cada turno, o jogador pode fazer ações como:

- Mover-se para uma casa;
- Saltar para uma casa;
- Atacar um monstro numa casa vizinha;
- Atacar alguém a distância;
- Usar bombas ou explosivos;
- Usar uma skill.

Quando o jogador faz uma ação, todos os inimigos fazem uma ação a seguir. Normalmente, existem vários tipos de inimigos que possuem várias características tais como:

- O tipo de ataque, por exemplo:
  - Corpo a corpo;
  - À distância;
  - Com explosivos.
- O tipo de movimento;
- A resistência.

# Implementação

A implementação do projeto será feita numa CGI em C, no sistema operativo Linux. Cada grupo tem liberdade na sua implementação do jogo. Em cada etapa, são avaliados vários itens sendo alguns obrigatórios (cada um deles valendo um ponto) e outros opcionais.

## Etapas

O projeto terá 3 etapas:

1. Etapa (23:59:59 do dia 09/04)
  - a. Criar tabuleiro de 10x10
  - b. Gerar aleatoriamente a posição dos obstáculos ( $\geq 20$ ) e dos monstros ( $\geq 10$ )
  - c. Respeitar o invariante de cada casa só ter um monstro/jogador ou um obstáculo
  - d. Mover o jogador
  - e. Criar novo nível quando se chega à saída
2. Etapa (23:59:59 do dia 21/05)
  - a. Obrigatório
    - i. Mover os monstros
    - ii. Matar monstros
    - iii. Matar jogador
    - iv. Usar ficheiros para guardar o estado
    - v. Calcular score e guardar um histórico dos melhores scores
  - b. Bónus
    - i. Pathfinding
    - ii. Ajudas, e.g.,
      1. Ao escolher um monstro, ilumina todas as casas que este ataca
      2. Opção que ilumina todas as casas atacadas por qualquer monstro
    - iii. Vários tipos de movimentos/ataques, e.g.,
      1. Ações com recursos escassos
      2. Movimentos de várias casas
      3. Ataques à distancia
      4. Ações que afetam várias casas
      5. Ações que mudam o estado do tabuleiro
    - iv. Manipulação de objetos (poções/armas/equipamento/feitiços)
    - v. Grelha hexagonal
    - vi. Usar cookies em alternativa ao ficheiro do estado
    - vii. Usar animações
3. Etapa (23:59:59 do dia 11/06)
  - a. Obrigatório
    - i. Documentação

- ii. Legibilidade
- iii. Não ter avisos ou erros quando o código é compilado com as seguintes opções: gcc -ansi -Wall -Wextra -pedantic -O2
- b. Bónus
  - i. Inimigos inteligentes, e.g.,
    1. arqueiros que escolhem posições de onde possam disparar contra o jogador
    2. Inimigos prevêm para onde o jogador vai e atuam de acordo com isso
    3. Cooperação entre inimigos (e.g., monstros que caçam em grupo)
  - ii. Jogador automático

## Avaliação

A aprovação do grupo implica cumprir pelo menos 10 dos 13 itens obrigatórios. A nota do grupo será dada pela avaliação dos itens obrigatórios e opcionais. A avaliação individual depende da prestação de cada elemento do grupo no projeto.