

Componentes da caverna:
* 1 jogador
* 1 Wumpus
* 1 ouro
* Entre 2 e 3 buracos

Especificações das salas:
* 16 salas (4x4)
* Tem fedor nas adjacentes v e h do Wumpus
* Tem brisa nas adjacentes v e h dos buracos
* Nunca tem ouro nas quinas da caverna
* Ouro, Wumpus e Buraco não ficam na mesma sala
* Herói sempre começa no 1.1

Representações das salas:
* Herói: P
* Wumpus: W
* Buraco: B
* Ouro: O
* Fedor: f
* Brisa: b

Prioridade:
O = B = W > P > f > b

Pontuações:
* Sair da sala com o ouro: +1000
* Cair em buraco ou ser morto pelo Wumpus: -1000
* Realizar um movimento: -15
* Usar a flecha: -100
* Matar o Wumpus: +500

Caverna
- mapa: Sala[4][4]
+ adicionarComponente(Componente c) + moverComponente(Componente c, char wasd) + removerComponente(Componente c) + toString() : String

Sala
- visivel : boolean - prioritario : Componente - restante : Componente[3]
+ tornarVisivel() + adicionarPrioritario() + getPrioritario() : Componente + adicionarComponente(Componente c) + compMaisImportante() : Componente + exporSala() + heroiSaiu() + toString() : String

Montador da Caverna
- mapainicial : String[]
+ gerarMapa(String[] mapa) : Caverna - gerarCaverna() : Caverna - gerarSalas(String[] mapa) : Sala[][]

Controle do Jogo
- heroi : Heroi - pontuacao : int - fase : caverna - scanner : Scanner
- imprimirInterface() - executarComando(String com) - moverHeroi()

AppMundoWumpus

CSV Handling
- String dataSource - String dataExport - String[][] commands - String[] state;
+ CSVHandling() + getDataSource() : String + setDataSource(String dataSource) + getDataExport() : String + setDataExport(String dataExport) + requestCommands() : String[][] + exportState(String[] state) - readCSV() - writeCSV()

Heroi
- nome : String - flecha: Flecha - bolso : Ouro
+ getFlecha() : Flecha + flechaEquipada() : boolean + getOuro() : Ouro + percepcao() : Componente[] + capturarOuro() : boolean + moverHeroi(char wasd) + disparar() + confrontarWumpus() : boolean + toString() : String

Ouro
+ setCoordenadas() + toString() : String

Talvez substituir por um método "serCapturado" que seta linha e coluna como null

Buraco
- jogador : Heroi
+ capturarHeroi(Heroi h) - gerarBrisa() + toString() : String

Wumpus
- gerarFedor() + confrontarHeroi(Heroi h) - dissiparFedor() - morrer() + toString() : String

Brisa
+ toString() : String

Fedor
+ toString() : String

Flecha
- equipada: boolean
+ equipar() + statusEquipada() : boolean

Componente
linhaSala : int # colunaSala : int # caverna : Caverna
+ getLinha() : int + getColuna() : int + posicionar()

