

Alunos: Gabriel Medrano Silva Ra: 247113
Henrique Pacheco Manoel Ra: 247165

Destaques de código:

```
public class Sala {  
    private Componente[] componentes = new Componente[4];  
    private boolean descoberta = false;  
    private int numComponentes = 0;  
  
    public Sala() {};
```

Gostaríamos de destacar a classe sala pois nessa utilizamos polimorfismo para criar um vetor de componentes mesmo estes sendo instanciados com tipos específicos de componentes. Permitindo-nos organizarmos nosso código de uma melhor maneira e como todos os componentes possuem toString pois este método é dado como abstrato podemos usá-lo para quaisquer componentes, tendo resultados diferentes para cada variação.

```
public Buraco(int x, int y) {  
    super(x, y);  
    Posicao adjacentes[] = posicao.adjacentes_validos();  
    for(int i = 0; i < 4; i++) {  
        if(adjacentes[i] != null) {  
            Componente brisa = new Brisa(adjacentes[i].getX(), adjacentes[i].getY());  
            caverna.adicionarComponente(adjacentes[i], brisa); // Adiciona na caverna as brisas  
        }  
    }  
}
```

Outro ponto a ser destacado é o uso do super na construção dos objetos permitindo que mudanças a serem feitas no construtor do componente serão aplicadas aos seus herdeiros de maneira automática ajudando assim em futuras expansões de código, vale ressaltar que nesse construtor específico assim como no Wumpus possuem uma propriedade especial que difere dos demais que é a geração de componentes adjacentes em suas construções.

```
Caverna caverna = new Caverna();  
caverna = MontadorCaverna.montar(arquivoCaverna, arquivoSaida, arquivoMovimentos);
```

Utilizamos esta parte do programa para destacar que o Montador Caverna se preocupa somente realizar sua função de verificar se a caverna segue os parâmetros impostos e criá-la, permitindo localizar possíveis erros de código, uma vez que somente essa parte monta a caverna, e melhorar a visualização e entendimento do erro.