

# Relatório - EP Fase 2

## Laboratório de Programação 2

28 de outubro de 2013

### 1 Integrantes

- Victor Sanches Portella - N° USP: 7991152
- Lucas Dário - N° USP: 7990940
- Ruan Costa - N° USP: 7990929

### 2 Introdução

Neste relatório temos como objetivo dar uma explicação concisa sobre a organização das classes no funcionamento do programa atualmente. Para isso, explicaremos o funcionamento do programa inteiro, e iremos nos aprofundar nos assuntos que forem necessários.

### 3 Funcionamento

Nessa fase o EP ainda não têm o funcionamento de um jogo em si, mas já é possível ver diversos robos agindo em paralelo. Para isso, o script em Perl<sup>1</sup> traduz diversos programas para um arquivo chamado **Main.java**.

Todos os robôs traduzidos serão colocados no mesmo time. Ao rodar o programa, os programas de cada robô será executado até o fim. Quando todos os robos tiverem acabado de executar, o programa será finalizado.

Cada robô imprime as instruções que executou para que seja possível que nós possamos acompanhar a execução de cada robô.

---

<sup>1</sup>Explicação do mesmo feita mais a frente

## 4 Classes

### 4.1 Classe Arena

Essa classe representa a Arena em si. Nela, temos os seguintes atributos:

- **Mapa**: Objeto do tipo **Mapa** que representa o mapa em si. Nesse objeto existe uma matriz de Terrenos, cada um contendo informações como robôs naquele terreno, assim como cristais, bases, etc. Melhores explicações sobre esse objeto serão dadas mais a frente.
- **List<Robos>**: Objeto do tipo **List** usado como uma lista de objetos do tipo **Robo**. Essa lista contém todos os robos em jogo.
- **Tempo**: Uma variável inteira representando o número de rodadas já passadas. O valor inicial dessa variável é 0 e, a cada rodada, é incrementada em 1.

Principais métodos dessa classe:

- **Arena(Mapa)**: Construtor da classe. Recebe uma mapa, e retorna uma arena vazia, ou seja, com nenhum robo, o tempo é iniciado com 0, e o mapa passado é definido como o mapa da arena.
- **void atualiza()**: Função que faz a Arena executar uma rodada. Atualmente, essa função percorre a lista de robôs de forma aleatória, executando em cada robô uma instrução de máquina. Em caso de conflito de instruções, tem preferência o robô que for executado primeiro naquela iteração.
- **void sistema(Operacao)**: Função chamada pelos robôs para fazerem chamadas ao sistema. Toda vez que os robôs tentam executar uma ação que influencia seu exterior, ele só pode fazê-lo chamando esse método. O objeto do tipo **Operacao** recebido descreve qual operacao está sendo executada, e qual objeto Programável está tentando executar aquela ação. Dependendo da operação, esse método mexe na pilha do robo, e executa as ações pedidas, caso possível.
- **void insereExercito(Programa[],Posicao[],int)**: Função para inserir um exército de robôs na Arena. Para isso, recebe três argumentos:
  - + **Programa[]**: Um vetor de objetos do tipo **Programa**. Para cada programa, será criado um robô a ser inserido na Arena.

- + **Posicao[ ]**: Um vetor com posições x,y indicando onde cada robô criado deve ser posicionado, seguindo a ordem do vetor de programas.
- + **int time**: Um inteiro que indica a qual time esse exército pertencerá.

## 4.2 Classe Programa

## 5 Interfaces

## 6 Tradutor em Perl