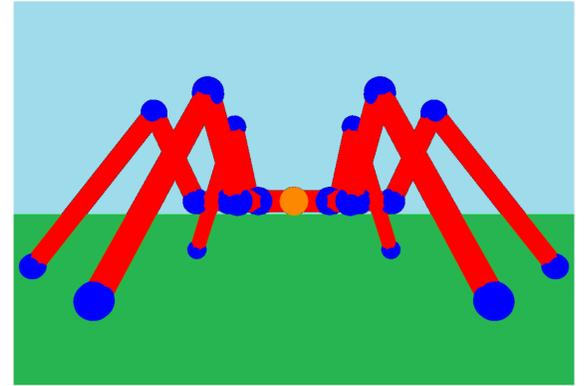


# Hexapod Battle

## - AI competition meets paintball

### Introduction

Hexapod Battle is an AI competition, in which two teams consisting of five players (hexapods) each, fight in a game of paintball against each other. The AIs must be written in Java and are restricted in their Java-Bytecode usage per frame. In contrast to just giving each robot a thread to run on, this ensures a fair and reproducible execution of the robots.



A hexapod; the orange sphere is used as head and gun of it

### Step 1: Familiarising with the Battlecode engine

Hexapod Battle uses the code of the MIT AI competition Battlecode, which manages the Bytecode-processing and also allows to modify Bytecode costs, so that functions (e.g. compute-intensive functions like sin, cos, sqrt, etc. ) can be treated as less expensive.

### Step 2: Game Logic

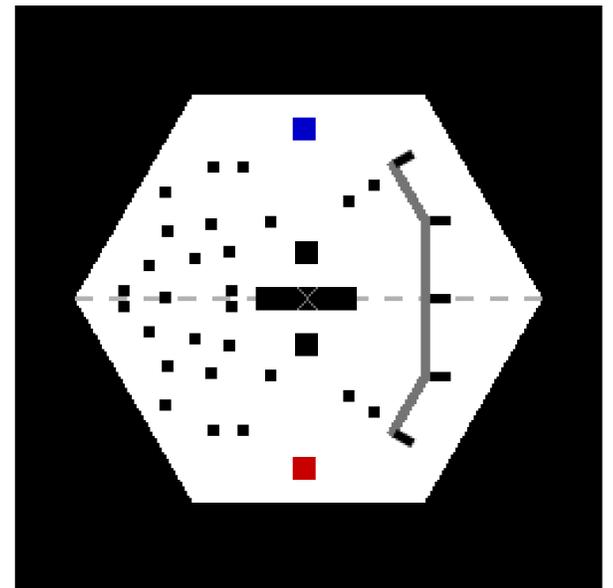
All the rules of the competition had to be implemented.

A match lasts either until one team reaches seven points or ten minutes have been played. A point is gained if one robot reaches the start area of the opposing team.

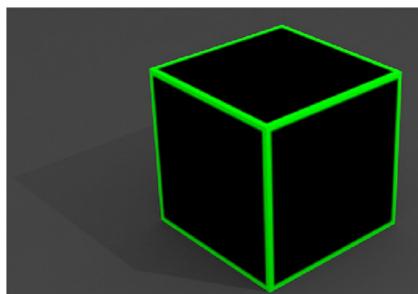
It was especially difficult to build everything suited for AIs instead of humans. So, now every hexapod can send out ten raycasts per frame from its head for visual input, additionally, hexapods can communicate through this way, but only 3 bit per frame.

Robots can shoot once every frame, but only have limited ammunition.

A robot which tries to do something which is not possible, has an error in its code or touches another robot or an obstacle explodes immediately.



Top view draft of a map in Hexapod Battle; black are bunkers; dark grey are flat bunkers and the coloured areas are the start areas of the hexapods. Reaching the opponent start area, wins you the current point



Tron-style objects

### Step 3: Rendering the competition

Hence the competition is only "played" by AIs, the visual part is only to watch them play. Here, a Tron-like style was used to make the very simple objects look more appealing.

The game logic and the representation are decoupled. So games are computed and then exported into a file which can be viewed separately.

### Plans for the future

Program a good AI and with that balance all the parameters.

Implementing different tournament types.

Building a map editor.

### Used technology

Hexapod Battle is programmed in Java, for the visual representation jME 3.1 is used.

The core of the code is from the MIT AI competition Battlecode.

### Related project

This project is related to **Hexapod Walk**, in which the walking animations were created.