

# 2D Strategy Simulator for 3D simulation

Jens Fischer

Hochschule Offenburg, Germany  
jfische2@stud.hs-offenburg.de

## Abstract

Due to non-determinism, limited control and an expensive physics simulation, the development of high-level behaviors and strategies in *rcssserver3d* is cumbersome.

To solve these issues, we created a greatly simplified 2D simulator for testing and debugging. It runs a full simulation with 22 instances of *magmaOffenburg*'s agent runtime, while giving much more control over the simulation, such as a pause functionality, the ability to change the simulation speed or even jumping back in time.

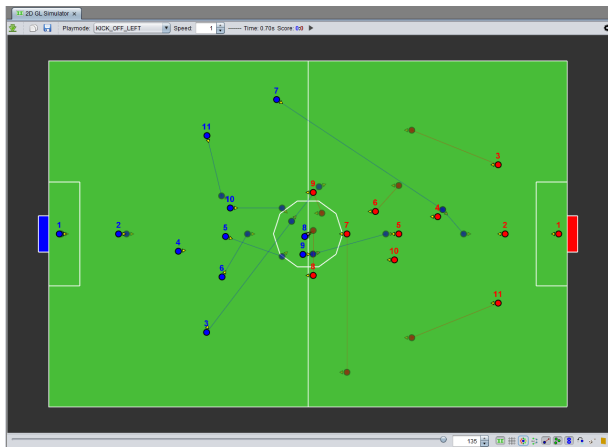


Figure 1: Visualization of the 2D Simulator in *magmaDeveloper*.

## 1 Approach

The 2D simulator is intended to behave similarly to *rcssserver3d* on a high level, while also getting rid of expensive physics and non-determinism that are largely detrimental when testing certain features. To achieve this, the simulation is limited to a few basic, high-level commands: *beam*, *walk*, *kick* and *dribble*. For kicks, we measured distance and angle distributions in *rcssserver3d* to achieve similar results. This kick noise is the only non-deterministic part of the simulation, and can be turned off or seeded.

In order for *magma*'s existing agent runtime to be used in the 2D simulator, we implemented an adapter that translates our behaviors to 2D simulator commands. Apart from that, very few adjustments to existing agent code were necessary. The simulator is written in Java, which allows a tight integration with our existing development tools, bundled as the *magmaDeveloper* application. In this case, we added a "Simulator" view to visualize the simulation's state (Figure 1).

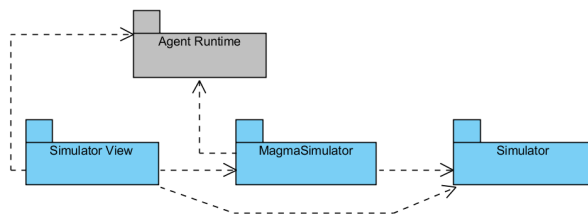


Figure 2: Relationships of the different components.

Architecture-wise, the *Simulator* is a separate component, independent of *magma*'s agent or UI logic (Figure 2). The aforementioned adapter between *Agent Runtime* and *Simulator* is implemented as so-called *MagmaSimulator* component. The *Simulator View* is the only component that depends on all three other components.

## 2 Results

Over the past half year, the simulator has already proven to be an invaluable tool for our development process. The user has much more control over the simulation than in *rcssserver3d*, the simulation can be paused, slowed down or sped up. Due to the simplicity of the simulation, to keep a complete copy of the simulation state at 50 cycles per second, allowing jumps to arbitrary points in time.

With only a few minor modifications, we were able to make the simulator usable for *Sweaty*, our robot in the Humanoid league. Many bugs that could only be found on real hardware before can now be solved more quickly by testing in the simulator first.