A New GameController

B-Human's Contribution to the Standard Platform League Open Challenge 2012

Motivation

The GameController is an important piece of infrastructure for our league. It has to be updated each year to reflect the corresponding rule changes. In recent years, it has been maintained by a small number of people from the Humanoid League and the SPL (including B-Human team members). Unfortunately, the old GameController does not possess a clean Model-View-Controller architecture, which made it increasingly difficult and error-prone to implement the newest rules. For instance, clicking a button not only directly updates the UDP packet sent to the robots, but it also directly rearranges elements of the graphical user interface, resulting in many dependencies between different actions. Another design choice is to store the team colors implicitly (the first team is always the blue one and it is shown on the left side), which, e.g., made it difficult to implement the new penalty shoot-out (sides are switched, colors are not). Finally, it does not measure time correctly on slower computers, because rather than using the system clock, it counts timer events, which, e.g., resulted in halves to last closer to eleven rather than ten minutes at the RoboCup German Open 2012 according to our video footage.

Some other RoboCup leagues use infrastructure competitions to motivate teams to implement the tools the leagues need. Since the SPL does not have such a competition, we present our proposal for a new GameController in the Open Challenge.

Implementation

The new GameController offers four important improvements over the old one: a clean Model-View-Controller architecture, a user interface that correlates with the rules rather than offering the options to somehow implement them, an additional keyboard interface, and the correct measurement of time.

The **architecture** is based on a strict separation between model, GUI, network interface, and actions. Actions only update the model and notify the graphical and network interfaces that something has changed. These will then update their state automatically based on the data in the model. This abstraction simplifies modifications a lot, allowing future rule changes to be implemented easily.

The **user interface** has a clear structure: one team on the left, the other one on the right, the game state and the penalties in the middle. On the one hand, the new GameController disables all actions that would not be legal at the current state of the game, such as "Ball Holding" in the Ready state. On the other hand it comes with extra functionality, for example with buttons for Global Game Stuck (with kick-off for the team whose button is clicked). This way the new GameController provides a legal state at any time. Even if an operator pushes a button by mistake, there is a timeline with an *Undo* functionality to get back to any previous legal state. To use the GameController really quick, inputs can be made via the **keyboard**.

Time is measured using time stamps rather than counting it. The actual durations are calculated from the current system time and the time stamps whenever they are needed. Therefore, the performance of the PC used for the GameController does not influence time measurements. This representation of time also supports the Undo feature: decisions can be undone, but the time continues to run.

The new GameController is written in Java, just as the old one. The network packets are still compatible to existing implementations such as the GameStateVisualizer, so switching to the new one is easy.

Demonstration

B-Human's Open Challenge demonstration will be the start of a game controlled by the new GameController while explaining its benefits and pointing out some of its key features.

We will publish the new GameController on GitHub to allow future Technical Committees and teams in general to participate in its further development.