Amsterdam Oxford Joint Rescue Forces: Community Contribution

May 17, 2010

The Amsterdam Oxford Joint Rescue Forces have made several contributions to the Virtual Robot Competition's new UT3-based USARSim simulator over the past year, and intend to make several more over the next couple of months. This document details those contributions.

1 Battery (Bas Terwijn, 7points, complete)

The Battery limits the time/power that a robot may use before becoming inactive. It is now part of the sensor class. The class has for instance the following methods:

`getCurrentEnergy`	:	report current energy of battery (in Joule)
`discharge`	:	discharge battery with energy (in Joule)
`expectedLifeTime`	:	expected lifetime in seconds based on exponential moving average of discharge

2 ComServer interface (Bas Terwijn, 7points, complete)

The ComServer interface provides the interface for the external Wireless Simulation Server tool. It listens permanently on port 7435 for requests from the external Wireless Simulation Server. The interface handles two commands:

`GETPOS`	:	Interface for DistanceOnlyPropagationModel (signal strength estimation based on distance only)
`GETOBS`	:	Interface for ObstaclePropagationModel (signal strength estimation based on distance and attenuation of obstacles)

3 Fire and Smoke (Okke Formsma, Nick Dijkshoorn, Sander van Noort, 10 points, complete)

Environmental model, including the scripts for the sensor response on these effects. Realistic simulation of fire and smoke requires a combination of flames, sparks and smoke emitters, as shown in figure 1. The emmitted particles should be detected by the sensors, as for instance the range scanner. The reaction of the range scanner on smoke is validated [1] on datasets made available by other research groups and by own measurements. This environmental model was for the first time used during the Teleoperation test at the Iran Open 2010[1].

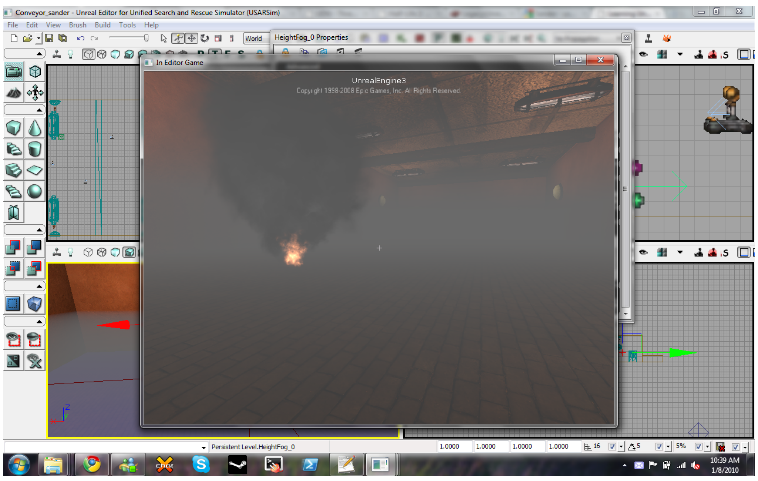

Figure 1: The combination of regional and local smoke in the Unreal Editor.

[1]Availabe at sourceforge: `http://usarsim.svn.sourceforge.net/viewvc/usarsim/Unpublished/USARSimMaps/DM-RobotTeleopRobrecht_v5.ut3`

4 Kenaf (Julian de Hoog, 10 points, in progress)

The Kenaf is a 6-track mobile robot platform which is designed for drastic performance gain of uneven terrain mobility. The robot can be controlled as previously by the DRIVE and MULTIDRIVE commands:

`DRIVE {Left a} {Right b}`

where a and b are the velocities (floats) of the Kenaf's main tracks (skid-steered)

`MULTIDRIVE {FRFlipper x} {FLFlipper y} {RRFlipper z} {RLFlipper r}`

where x, y, z, r are the flipper angles in radians (front right, front left, rear right, rear left respectively).

Modeling and programming of the Kenaf is still in progress, but we hope to have it completed early June 2010.

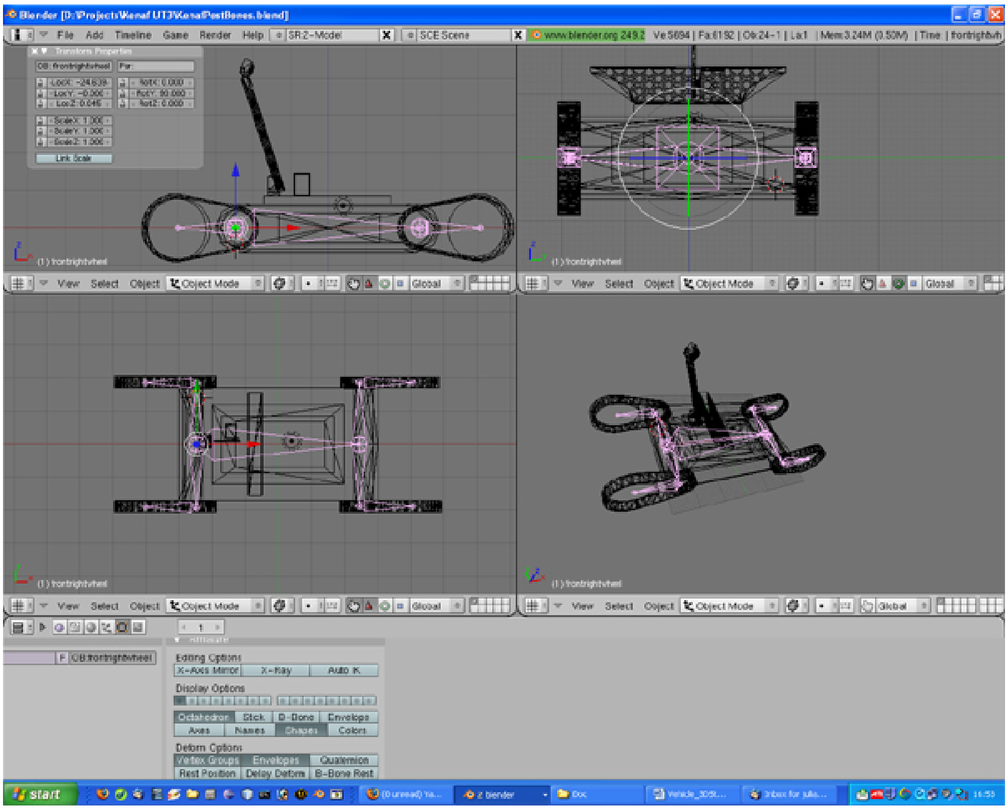

Figure 2: An 3D model of the Kenaf have been created in Blender.

The 3D model of the Kenaf (with the right bones) has been created in Blender 2 and exported to Unreal with a psk/psa script[2]. PSK is the acronyme of the file format for the skeleton mesh and PSA for the animation sets. Initially the scaling was not correct (the Kenaf too small), but rescaling of the core model gave dimensions close the expected 57cm x 40cm x 30cm.

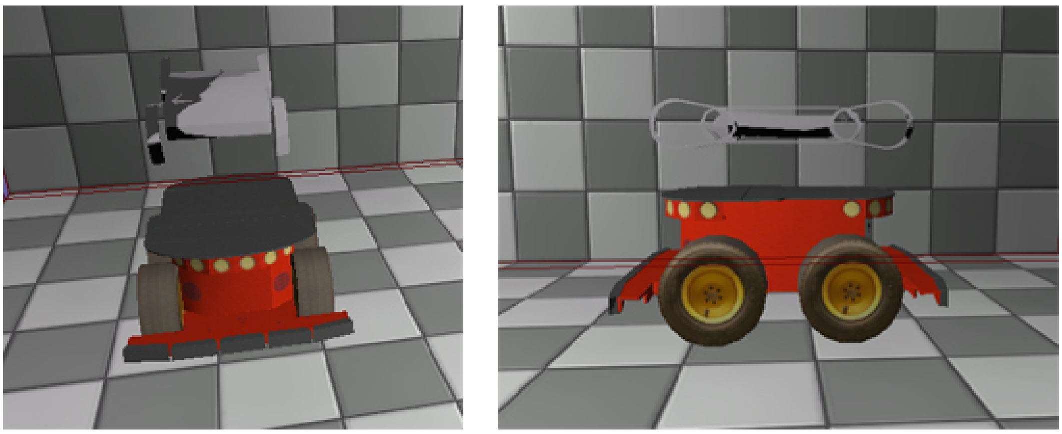

Figure 3: The Kenaf inside UT3 compared with the dimensions of the P3AT (50cm x 49cm x 26 cm).

Currently, the code from the native Goliath tank model is modified to model the behavior of the track based Kenaf properly (the Goliath-model seems to be how other custom built track-based vehicles have been developed).

[2]Availabe at Google-code: `http://code.google.com/p/unrealtacticalmod/`

5 Radar sensor (Niels Out, 10 points, started)

A sensor that is not part of USARsim (both UT2004 and UT3) is radar. Radar only recently became affordable and lightweight. Radar has several advantages over other sensors, such as the capability to see the reflections of multiple obstacles.

Figure 4: A map generated in an outdoor environment under good circumstances (laser measurements in black, radar measurements in pink).

The implementation should correctly model the response of the sensor on obstacles. There is an extensive data set available[3] to validate this behavior (even under extreme circumstances). The advantages of the new sensor can be demonstrated by creating a map (see Fig. 4) under circumstances which are difficult for other sensors.

6 Victim behavior (Chaim Bastiaan, 10 points, started)

Unfortunatelly at a disaster victim could be present, and finding those victims should have high priority for a robot team. An important feature of living victims that they can breath, move and make noise.

To be able to demonstrate advanced victim detection algorithms the victims inside UT3 should have dynamic behavior. Currently, the research focused on the example models provided with the Unreal Developer Network[4]. These models are designed to be realistic proportioned (see Fig. 5), have moving eyes, lip-synching abilities and sign-language capable hands.

Figure 5: A realistic male and female model designed by Tom Lin.

Currently, the models are successfully imported in the Unreal Development Kit[5]. The collision frames of the models needed some tuning, but this issue is solved. More important is that the models can show movements and behaviors through triggered actions.

References

[1] O. Formsma, N. Dijkshoorn, S. van Noort and A. Visser, “Realistic Simulation of Laser Range Finder Behavior in a Smoky Environment”, in “Proceedings CD of the 14th RoboCup International Symposium”, June 2010, to be published in ”RoboCup 2010: Robot Soccer World Cup XIV”, Lecture Notes on Artificial Intelligence series, Springer, Heidelberg.

[3] Courtesy of the Australian Centre for Field Robotics \http://sdi.acfr.usyd.edu.au/

[4] http://udn.epicgames.com/Two/UnrealDemoModels.html

[5] http://www.udk.com/