

rFactor Internals Plugin

www.rfactor.net



Overview

The enhanced internals plugin architecture, incorporated in rFactor version 1.070, exists to expose internal simulation data to 3rd party developers. This architecture can be used for many different applications such as telemetry programs, motion bases, head trackers, and statistical databases. Custom hardware controls for certain simulation dynamics as well as custom force feedback routines can also be created using this system.

This document is written under the assumption that the reader has an intermediate knowledge of C++ and object orientated design.

Section 1 Running the Example Plugin.

Copy the plugin to the \rFactor\Plugins directory. A compiled version can be found in the \ExamplePlugin\VC7\Release directory. Run rF Config.exe and make sure you choose a resolution that is less than your current desktop resolution and ensure the check box for windowed mode is checked. Then start rFactor and enter a race. Once the track has loaded click on the drive button. You will see the console this example plugin uses come alive. There will also be a text file in the root directory of the simulation that has captured telemetry data. If you see this item then you have successfully installed the example plugin.

Section 2 Soft real time programming.

A perfect example of soft real time programming are video games. Since rFactor fits in this genre it has a soft real time requirement. What this basically means is that code you insert into rFactor via the internals plugin interface must be efficient. For example if rFactor is running at 100 frames per second (FPS) out of the box, and code added via the plug in takes 1 msec to run, then the new code in the plugin has reduced the FPS to ~ 91 FPS. It should be obvious that a poorly written plugin could adversely effect the FPS in rFactor. The internals plugin interface should be used for reading data and having that data sent to some other application or device. However, if an algorithm is needed to format or calculate data, one could be placed in the plugin if it is properly designed and tested. If you are releasing this plugin to the public, remember rFactor's minimum hardware specifications. Those specifications can be found here: <http://www.rfactor.net/index.php?page=features#sysreqs>. Also be aware that rFactor attempts to update telemetry at 40 FPS when the vehicle is under AI control and 90 FPS when the player is in control of the car. Conditions may exist that could reduce these targeted frame rates.

Section 3 Telemetry data

The example included with this readme gives a developer an example of how to read data from the Telemetry class in InternalsPlugin.hpp. This class is the source of all telemetry data. The data members of this class are refreshed every time a graphics frame is produced but no faster than 90 FPS. The interface for this class is exposed and is commented in the included source files.

Section 4 Graphics data

Certain graphics data can be read using the GraphicsInfo structure in InternalsPlugin.hpp.

Section 5 Hardware input

Hardware input can be used to control certain aspects of the simulation. Using data from a hardware source you can edit these controls:

ToggleAIControl	DriverHotSwap	PassengerSelect
DisplayMode	PitRequest	PitMenuUp
PitMenuDown	PitMenuIncrementValue	PitMenuDecrementValue
LookLeft	LookRight	RearLook
DisplayVehicleLabels	InstantReplay	Pause
RestartRace	ToggleFreeLook	ZeroFreeLook
Horn	Headlights	LookUp
LookDown	LookRollLeft	LookRollRight
AdjustSeatFore	AdjustSeatAft	AdjustSeatUp
AdjustSeatDown	ToggleMirror	ToggleHUDStats
ToggleHUDTach	ToggleHUDMFD	ToggleOverlays
RealtimeChat	QuickChat01	QuickChat02
QuickChat03	QuickChat04	QuickChat05
QuickChat06	QuickChat07	QuickChat08
QuickChat09	QuickChat10	QuickChat11
QuickChat12	Screenshot	TimeAcceleration
SteeringHelp	OppositeLock	BrakingHelp
StabilityControl	SpinRecovery	Invulnerability
AutoShifting	TractionControl	AntiLockBrakes
AutoPit	AutoClutch	SharedMemUp
SharedMemDown	SharedMemLeft	SharedMemRight
SharedMemSelect	SharedMemCancel	ViewNextVehicle
ViewPrevVehicle	ViewMyVehicle	CycleDriving
CycleOnboard	CycleSwingman	CycleSpectator
CycleTracking	SwingmanPitchUp	SwingmanPitchDown
SwingmanYawLeft	SwingmanYawRight	SwingmanDecRadius
SwingmanIncRadius	SwingmanReset	Camera_CamEd_Slow
Camera_MoveForward	Camera_MoveBackward	Camera_MoveLeft
Camera_MoveRight	Camera_MoveUp	Camera_MoveDown
Camera_PitchUp	Camera_PitchDown	Camera_RollLeft
Camera_RollRight	Camera_YawLeft	Camera_YawRight
Camera_ZoomIn	Camera_ZoomOut	Toggle_HWPlugin
VoiceChat_PTT	CameraChange	ResetFFB

In Example.cpp you will see this method `bool ExampleInternalsPlugin::CheckHWControl(const char * const controlName, float &retVal)`. The editable control name, given above, is the first parameter and the second parameter is the value the control will be set to.

Section 4 Force feed back

The method `bool ExampleInternalsPlugin::ForceFeedback(float &forceValue)` has an incoming value that the simulation would normally use. You can either modify it or replace it with a value of your choosing.

All information, documentation, or software ("Information") is being provided "AS IS". Image Space Incorporated does not and cannot warrant this Information (including any fixes and updates) or the performance or results obtained by using it. Image Space Incorporated makes no warranties of any kind, either expressed or implied, including but not limited to, non-infringement of third party rights, merchantability, or fitness for a particular purpose. To the extent you use or implement this Information, you do so at your own risk. In no event will ISI be liable to you or others for any damages arising from your use or your inability to use any of the information, documentation, or software. Commercial exploitation is prohibited. Image Space Incorporated may terminate, change, or suspend any aspect of this Site and the contents provided at any time. Please visit <http://www.rFactor.net/legal.html> for additional or updated information.