

# Zeitnahme für A – B Strecken bei GTR2 / GTL / rFactor

## Tutorial by renntom

Für dieses Tutorial benötigte Programme: - BTB-Pro  
(je nach SIM und Vorgehensweise, werden - BTB-Evo  
nicht alle Prog. benötigt) - 3DsimED  
- GEditor

### Zunächst einmal zur Arbeitsweise und zum Verständnis:

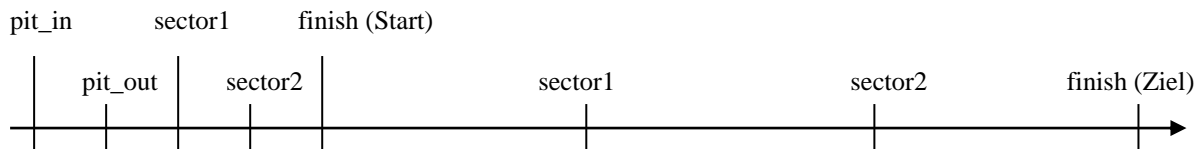
GTR2 / GTL und rFactor sind für Rundstreckenrennen gedacht, und somit ist die Zeitnahme natürlich auch anders wie bei A-B Prüfungen. Es gibt, nicht wie z.B. bei Rallye Spielen, eine Start- und Ziel-Lichtschranke, sondern nur eine Ziel-Lichtschranke. Im Spiel sind diese „Lichtschranken“ sogenannte „Time-Gates“.

Unser Ziel ist es also, dem Spiel vorzumachen, dass wir, obwohl wir nur eine Strecke fahren, eine ganze Runde gefahren sind.

verwendete Time-Gates: - finish  
- sector1  
- sector2  
- pit\_in  
- pit\_out

Für die eigentliche Zeitnahme benötigen wir *finish*, *sector1* und *sector2*. Die *sector-Gates* sind nicht nur für die Zwischenzeiten, sondern auch zur Kontrolle da, ob die Runde auch auf vorgeschriebenem Weg befahren wurde, d.h. erst wenn die *sector-Gates* in der richtigen Reihenfolge passiert wurden, erfolgt beim Überfahren des *finish-Gates* eine Zeitnahme.

Damit die Zeitnahme also Funktioniert, müssen die Gates folgendermaßen angeordnet werden:



Wie man an der Grafik erkennen kann, ist der Trick dabei, die Rundstrecken-Zeitnahme zu imitieren, daß die *sector-* und *finish-Gates* doppelt vorhanden sind.

**ACHTUNG:** Ich empfehle die Zeitnahme erst ganz am Ende eurer Arbeit zu machen, da sich beim Exportieren der Strecke außer den Time-Gates nichts mehr ändern sollte.

### doppelte Time-Gates erzeugen:

#### Möglichkeit 1:

In BTB zuerst einmal alle benötigten Time-Gates, also *pit\_in*, *pit\_out*, *sector1*, *sector2* und *finish* so setzen, daß nach verlassen der Box (*pit\_out*), zuerst *sector1* dann *sector2* und dann *finish* passiert wird, wobei der Abstand zwischen den Gates mind. 3-4 Wagenlängen betragen sollte, und das *finish-Gate* an der Startlinie platziert ist. Jetzt die Strecke exportieren und die dabei entstandenen *xsector1.gmt*, *xsector2.gmt* und *xfinish.gmt* Dateien am besten in einen dafür angelegten Ordner abspeichern.

Jetzt setzen wir die Gates neu, sodass die *sector-Gates* auf der Strecke, dort wo die Sektorzeiten genommen werden sollen, platziert werden und das *finish-Gate* die Ziellinie markiert. (*pit\_in* und *pit\_out* ändern sich nicht) Jetzt erneut exportieren und die nun entstandenen *xsector1.gmt*, *xsector2.gmt* und *xfinish.gmt* Dateien umbenennen in: *xsector01.gmt*, *xsector02.gmt* und *xfinish01.gmt* und in den gleichen Ordner, wie die beim 1. exportieren entstandenen Dateien, kopieren.

#### Möglichkeit 2:

In BTB zuerst einmal alle benötigten Time-Gates, also *pit\_in*, *pit\_out*, *sector1*, *sector2* und *finish* so setzen, daß nach verlassen der Box (*pit\_out*), zuerst *sector1* dann *sector2* und dann *finish* passiert wird, wobei der Abstand zwischen den Gates mind. 3-4 Wagenlängen betragen sollte, und das *finish-Gate* an der Startlinie platziert ist. Jetzt die Strecke in 3DsimED öffnen, die benötigten Gates kopieren, neu setzen und dann die .gmt Dateien in einen dafür angelegten Ordner exportieren (save model data). Dabei werden alle .gmt Dateien exportiert und es entstehen automatisch die 6 benötigten Time-Gate Dateien.

## Time-Gates ins Spiel bringen:

Damit die Time-Gates nun auch richtig funktionieren müssen sie in die dafür vorgesehenen Dateien:

GTR2: *trackname.gtr*

GTL: *trackname.gtl*

rFactor: *trackname.mas* (nur wenn vorhanden)  
eingefügt werden.

Ab hier unterscheiden sich die Spiele in der Art, wie sie die Time-Gates verarbeiten.

### GTR2 / GTL:

Da beide Spiele nur jeweils eine .gmt Datei verarbeiten können, müssen wir die vorher erzeugten Dateien zusammenfügen.

Dafür öffnen wir z.B. die *xsector1.gmt* und die *xsector01.gmt* Dateien in 3DsimED (import). Jetzt kopieren wir die *xsector01.gmt* (strg + c), fügen diese in die *xsector1.gmt* (strg + v) ein (bei der Nachfrage Update drücken) und exportieren diese dann als *xsector1.gmt* (Save Model Data). Wenn man die ursprüngliche Datei nicht überschreiben will, sollte die neue Datei in einen neuen Ordner (sinnvoll für nachfolgende Möglichkeit 1), oder in den, von BTB erzeugten, vorhandenen *mas* Ordner (sinnvoll für nachfolgende Möglichkeit 2), gespeichert werden. Das gleiche machen wir mit den anderen Dateien auch.

Jetzt haben wir also 3 .gmt Dateien, *xsector1.gmt*, *xsector2.gmt* und *xfinish.gmt*, die jeweils 2 Time-Gates enthalten. Diese müssen wir nun in die dafür vorgesehenen Dateien: *trackname.gtr* bzw. *trackname.gtl* einfügen. Auch hierfür gibt es wieder 2 Möglichkeiten.

### Möglichkeit 1:

Wir öffnen die *trackname.gtr* bzw. *trackname.gtl* mit dem GEditor und importieren zuerst die von uns erzeugten *xsector1.gmt*, *xsector2.gmt* und *xfinish.gmt*. Diese sind nun jeweils 2 mal vorhanden. Jetzt erst kann man die ursprünglichen löschen und dann die Datei speichern (Save GTR File). Die Datei Schließen und dann erneut öffnen, den 1.Schritt wiederholen, also die 3 .gmt Dateien erneut importieren und dann nochmals speichern.

### Möglichkeit 2:

Wir kopieren die von uns erzeugten .gmt Dateien in den, von BTB erzeugten *mas* Ordner, in dem dann alle .gmt liegen und überschreiben die alten Dateien. Jetzt packen wir, mit Hilfe von 3DsimED alle .gmt Dateien zu einer neuen *trackname.gtl* bzw. *trackname.gtr* und überschreiben die vorhandene.

### rFactor:

Da dieses Spiel jeweils mehrere Time-Gates verarbeiten kann, können wir uns das Zusammenfügen ersparen.

Wir können direkt mit den 6 erzeugten .gmt Dateien arbeiten. Dazu ist es aber notwendig dem Spiel auch mitzuteilen, dass jetzt auch die neuen Time-Gates verarbeitet werden müssen.

Dazu öffnen wir die *trackname.scn* Datei und fügen einen zusätzlichen Aufruf für die neuen Dateien ein.

So sollte es dann aussehen :

```
Instance=xfinish
{
  Render=False
  Change=False
  MeshFile=xfinish.gmt CollTarget=True HATTTarget=False ShadowReceiver=False
  MeshFile=xfinish01.gmt CollTarget=True HATTTarget=False ShadowReceiver=False
  Response=VEHICLE,TIMING
}
Instance=xsector1
{
  Render=False
  Change=False
  MeshFile=xsector1.gmt CollTarget=True HATTTarget=False ShadowReceiver=False
  MeshFile=xsector01.gmt CollTarget=True HATTTarget=False ShadowReceiver=False
  Response=VEHICLE,TIMING
}
Instance=xsector2
{
  Render=False
  Change=False
  MeshFile=xsector2.gmt CollTarget=True HATTTarget=False ShadowReceiver=False
  MeshFile=xsector02.gmt CollTarget=True HATTTarget=False ShadowReceiver=False
  Response=VEHICLE,TIMING
}
```

Die Änderungen Abspeichern. **FERTIG !!!**

## Noch ein Tipp:

Dem aufmerksamen Leser ist wohl oben bei der Aufzählung der Dateien bei der rFactor Datei *trackname.mas* die Bemerkung (*nur wenn vorhanden*) aufgefallen. Anders als bei GTR2 und GTL erzeugt BTB-Pro beim Export von rFactor keine gepackten .mas Dateien, sondern legt alle von der Strecke erzeugten .gmt, .dds, .jpg usw. in den Hauptordner der Strecke. Da es sich hier um hunderte von Dateien handelt ist das natürlich sehr unübersichtlich. Deshalb empfehle ich dringend, mit Hilfe von 3DSimED alle .gmt Dateien in eine *trackname.mas* und alle Texturdateien in eine *trackname\_MAPS.mas* zu packen (die komprimierten kann man löschen). Achtung: dann natürlich nicht vergessen in der .scn Datei die Pfade und Ordner anzupassen.

In GTR2 und GTL werden diese Dateien nicht nur automatisch beim Export gepackt, sondern auch noch zusätzlich in 2 Ordner *mas* für .gmt und *tex* für Texturen abgelegt. Diese werden für das Spiel selbst nicht benötigt, und können gelöscht werden.

Ich hoffe ich habe soweit an alles gedacht und ausführlich genug beschrieben. Falls Ihr Fragen und Anregungen zu diesem Tutorial habt, schickt sie mir einfach.

Viel Spass und Erfolg bei eurer Arbeit !

renntom

Email: [info@toms-sim-side.de](mailto:info@toms-sim-side.de)