

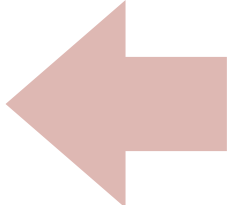
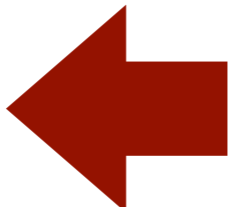
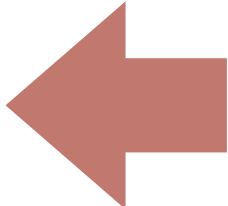
Seminar & Praktikum Computergrafische Systeme WS 2006/2007

Random Racer

Torsten Becker, Frederik Rudeck, Robert Timm

betreut durch Prof. Dr. Döllner, Stefan Maaß

Wo sind wir?

- Milestone 1
 - OS-X Port
 - Einarbeitung VRS und ODE 
- Milestone 2
 - Test-Code 
 - Design
- Milestone 3
 - Implementierung - Must Have 
 - Review
- Milestone 4
 - Implementierung - Should Have

- VRS läuft auf Mac OS X PPC+x86
- Bewegen des Fahrzeugs mit Hilfe der Tastatur
- Physikalisches Modell des Fahrzeugs
(Fahren durch Drehen der Räder und Lenkung durch Schwenken der Aufhängung)
 - ➔ Proof-of-Concept Implementation des ODE-Bindings
- eigene PoC-NURBS-Implementierung

- Design und teilweise Implementierung für Terrain Generierung
 - Nachladen bei Kollision mit Plane
 - Loader & Generator (Multithreaded)
 - Heightmaps/NURBS-Mischung
- In-Game Konsole mit History und Callback-Registrierung

Was kommt als Nächstes?

- VRSODE:
 - Refaktorisierung des Physik-Bindings um eine wiederverwendbare Schnittstelle zu haben
- Terrain Generierung
 - NURBS Cleanups
 - Loader, Generator und NURBS zusammenführen
- Random Racer
 - VRSODE, Terrain Generierung und Glue-Code kombinieren
 - Should Have und Nice to Have implementieren

- **InputHandling:**
 - Problem: keine zuverlässige Input Event Generierung
 - Lösung: SDL als Input Lib?
- **NURBS Implementierung:**
 - Problem: zwar direkt in OpenGL implementiert, dadurch aber zu sehr gekapselt für unsere Bedürfnisse
 - Lösung: eigene NURBS Implementierung
- **Erben von VRS::SharedObj**
 - Problem: in der falschen Reihenfolge erben bringt lustige Effekte
 - Lösung: in der richtigen Reihenfolge erben

