

UNIVERSITY OF CALGARY  
FACULTY OF SCIENCE  
DEPARTMENT OF COMPUTER SCIENCE



## MILESTONE 3 README

March 8, 2010

Michael Blackadar

Tom Flanagan

Don Ha

Ole Rehmsen

John Stuart

## 1 Introduction

This document is meant to provide some information related to the release of milestone 3 of the space racing simulation *eRacerX* by *Wallsocket Studios*. Section 2 points out major changes in the design as compared to the design document shipped with milestone 1 and the readme for milestone 2. Section 3 discusses as selection of new features in milestone 3. Section 5 outlines our plans for the milestone 4.

General information that has been revised since the readme for milestone 2 such as the controls (Section A), building from source and running the game (Section B) has been moved to the appendix at the end of the document.

## 2 Design decisions

In this section we will highlight some deviations from the design document and the schedule distributed with milestone 1 and the readme distributed with milestone 2 of *eRacerX*.

### 2.1 Spline Track

After experiencing many problems with creating a smooth track that negatively affected both driving stability and rendering, we decided to model the track as a B-Spline curve with a track profile cruve. We currently use a subdivision algorithm to compute a closed curve and interpolate the normals at each point. We are, however, planning to extend the concept to open b-spline curves to allow for discontinuities and jumps in the track.

### 2.2 Car physics using multiple capsules instead of a simple box

To achieve a better fit and to make collisions with the wall less punishing we have replaced the simple box representing the shape of the car in milestone 2 by a composition of multiple capsules.

## 3 New features

Here is a list of some of the new features since milestone 2 that in our opinion deserve explicit mentioning.

### **3.1 Variable gravity direction**

As planned, we have removed the global gravitational force and replaced it by an attractive force applied normal to the closest track segment. This allows for a very flexible track design with upside-down sections. The magnetism currently only acts on cars - meteors for example are not affected by it.

### **3.2 Improved camera**

To achieve a better sense of speed and handle the upside-down segments of the track better, we have changed the following camera to be mounted lower and focus on the road before the car instead of the car. We further changed it to use the road normal instead of a global normal as up vector but have it lagging behind enough to give a good feeling of the banking of the track.

### **3.3 AI**

Milestone 3 features a basic AI that is able to race along the track. It uses waypoints extracted from the b-spline underlying the track and is able to back up in case it gets stuck.

### **3.4 Meteors**

Also new are meteors that roam the skies around the track and potentially collide with each other, the track or the cars. The meteors can be divided into two categories: Random meteors that are relatively large, slow and cross an imaginary sphere around the whole track, getting respawned at a random location on the surface of the sphere as they leave the sphere in any direction. These meteors are purposely not spawned only around the player camera, as they may change the course of a race and for multiplayer should depend on where the camera is looking.

The second category of meteors is targeted meteors. Those are relatively small meteors that are aimed in front of the player car(s) and spawned every couple of seconds to force the player to react and change his route to avoid them.

### 3.5 3D sound

We have reworked the 3D sound code and finally actually put in some sound. The engine noise is pitching with the motor speed and adds to the realism of the game.

## 4 Known issues

### 4.1 Alpha blending order

The alpha blending for the blob shadow is not working properly if one flies off the track because the rendering is not happening in the correct order. This will be fixed by the next milestone.

### 4.2 Object Culling

Culling is currently disabled because there is a bug in the computation of the camera planes.

### 4.3 Fallback font

On some computers, loading our custom font causes a out of memory error (for unknown reasons). In this case we fall back to a system font.

## 5 Future work

In this section we want to touch on a few topics that we want to work on until the next milestone.

### 5.1 Smarter AI

We want to improve the AI to be able to avoid obstacles and each other. We also want it to behave less predictable and more dynamic. Lastly it should use boost to its advantage.

### 5.2 Improved boost

The boost as it is implemented currently is very simple and unlimited. For the next milestone we want to have a limited amount of boost that can be restored using the mechanism described

in the design document shipped with milestone 1. The amount of boost available should be displayed as part of the HUD.

### **5.3 Refined track**

We currently have two tracks, one relatively small, the other long but a bit unrefined. The latter needs to be improved to make it less likely to fall off and maybe tighten the turns a bit to make it more exciting.

### **5.4 Props**

The game still does not feel as fast as we would like it to be. One cause of this is that we have virtually no static objects that the player can relate to to estimate size and speed. For the next milestone, we want to have props along the roadside that will give you a better feeling for the speed the car is moving with.

### **5.5 Transparent track texture**

We want to use alpha blending to make the track partially transparent. With the solid texture used currently, a large part of the screen is taken up by the track, hiding the space with meteors, planets and nebulas. Being able to see through the track will make the space scenario much more fun and rewarding.

## **A Controls**

In this section we list the controls for our game. There are different controls depending on whether you are in the actual game or in a menu.

### **A.1 In-game controls**

Table 1: In-game controls

Command	Keyboard	Gamepad
Accelerate	W	Right Trigger
Reverse	S	Left Trigger
Break	Left Shift	B
Steer left	A	Left Analog Stick
Steer right	D	
Boost	Space	A
Cycle cameras	C	Y
Move flying camera	Arrow keys	
Reload tuning parameters	R	
Pause menu	Esc	Start

## A.2 Menu controls

Table 2: In-game controls

Command	Keyboard	Gamepad
Menu up	Up	Left Analog Stick
Menu down	Down	
Select menu item	Enter	A
Go back	Esc	B

## **B Building from source**

### **B.1 Prerequisites**

The following must be installed:

- Microsoft Visual Studio 2008
- Microsoft DirectX SDK (August 2009)
- NVIDIA PhysX SDK v2.8.1
- Python 2.6
- SWIG-1.3.40

### **B.2 Environment variables**

The following environment variables must be set (your paths may vary):

- DXSDK\_DIR = F:\Program Files (x86)\Microsoft DirectX SDK (August 2009)
- PHYSX\_DIR = F:\Program Files (x86)\NVIDIA Corporation\NVIDIA PhysX SDK\v2.8.1
- PYTHON\_DIR = F:\Python26\
- SWIG\_DIR = P:\swigwin
- 

### **B.3 Building**

Build the project eRacer/eRacer.sln.

### **B.4 Running**

To run the program, run `eRacer/run-release.py` or `eRacer/run-debug.py` with python.