UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

Wall Socket Studios presents

eRacerX™

MILESTONE 4 README

March 29, 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 4 of the space racing simulation *eRacerX* by *Wallsocket Studios*. Section 2 discusses a selection of new features in milestone 4. Section 3 lists some bugs that we have encountered and were not able to fix on time. Section 4 outlines our plans for final product.

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 New features

Here is a list of some of the new features since milestone 3 that in our opinion deserve explicit mentioning.

2.1 Boost

Boost will now not only let you go faster, it will also give you a small initial hop that can be used to clear meteors. The star effect has been improved for boosting to make for a better sense of speed. Boost also recharges significantly slower since the players additionally have the opportunity to steal boost.

2.1.1 Boost stealing

Players can now steal each others boost by driving up behind another vehicle. The stealing is automatic if one is close enough. A simple laser beam indicates whether or not you are currently stealing. One can steal until the other player has no boost left to steal.

2.2 Multiplayer

eRacerX now features a split-screen multiplayer. There can be 1,2 or 4 human players which can chose from one of 2 keyboard mappings and 4 controllers in the race setup menu.

2.3 Advanced menu

There is now a more advanced menu that lets you chose the color and controls of the players as well as the number of AIs to use. There is also the option to go back to main menu from paused state or even to restart the race with the same settings as the current race.

2.4 Hud

We have added a number of hud elements to display information about the game: In addition to the lap counter, the players position is displayed. There is a simple boost bar showing how much boost fuel is available and interface element showing how far each vehicle has proceeded in the lap. There is an indicator if the vehicle is facing the wrong way. There is also a personal finish screen to inform players in multiplayer that they have finished.

2.5 More content

To offer a more immersive experience, we have added additional sounds for meteor collisions as well as in the menu. Textures have been improved to have a grungier appearance and more car colors are available. Sections of the tracks are now surrounded by glass rings to provide more variety and a reference point that improves the sense of speed.

2.6 Driving code in C++

Porting the vehicle to C++ led to a more stable integration of the physics and thereby allowed us to further speed up vehicles.

2.7 Improved AI

The AI now dodges meteors and vehicles instead just driving straight and following the race line. This is especially noticable at the start where they used to drive through vehicles in their way. They also use boost whenever they are on a straight section of the track and are not running danger of jumping off the edge.

2.8 Improved track

We have modified the second track by compressing the exciting features on a shorter overall lap length to make driving less monotone.

3 Known issues

3.1 Support for variable screen resolution and aspect ratios

Currently, most of the hud and menu rendering depends on the resolution of the rendering surface. This is noticable if one plays two player multiplayer, where the aspect ratio is 8:3 instead of 4:3. Also the current resolution of 800 by 600 is relatively low and should be selectable from the menu instead. Support for full screen rendering would also be desirable. This has not been fixed yet due to a driver bug in DirectX, that results in sprites being rendered in the viewport on most graphics cards but not in others (such as the nVidia 8600M GT).

3.2 Sound bug

On some machines we had issues with sound popping, but not on others. This does not seem to correlate with the overall power of the machines or the frame rate. We will have to explore this in future.

3.3 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 in final version of the game (promise:-).

4 Future work

In this section we want to touch on a few topics that we want to work on for the final product. The last weeks will be used predominantly for bug fixing, beautification and game play tweaks.

4.1 Beautification

We want to have more props in the game to make the world look more vivid. The graphics of the hud will be revised. We want to add more visual effects, such as the partially transparent track (depends on alpha sorting), an animated boost stealing beam and particle effects for both the rocket propulsion and when the vehicle collides with the walls.

4.2 Gameplay tweaks

4.2.1 Even smarter AI

The AI should be less predictable and more dynamic. It should also use some rubber banding to ensure it does not get too far ahead, or behind. The AI also currently does not actually try to steal the players boost. This can be improved.

4.2.2 Discontinuous track and jumps

At this point a track is a closed b-spline. We would like to experiment with multiple open b-spline sections that would allow us to have jumps. This will only be used if it turns out to make the game more fun.

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	Keyboard1	Keyboard2	Gamepad
Accelerate	W	Up	Right Trigger
Reverse	S	Down	Left Trigger

Continued on Next Page…

Table 1 – Continued

Command	Keyboard1	Keyboard2	Gamepad
Break	Left Shift	Right Control	B
Steer left	A	Left	Left Analog Stick
Steer right	D	Right	
Boost	Space	Numpad 0	A
Reset	T	Right Shift	-
Cycle cameras	C	Numpad Decimal	Y
Pause menu	Esc	Pause key	Start

A.2 Menu controls

Table 2: Menu 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.