



Interim Report

Game Programming Lab 2010

Thomas Siegrist (ts)
David Gerhard (dg)
Philipp Keller (ph)
Jonas Hauenstein (jh)



1 Contents

1	Contents	2
2	Development state	3
2.1	Progress	3
3	Details on some targets	3
3.1	The Physics	3
3.2	Instancing	3
3.3	Deferred rendering	4
3.4	AI for cops and civilian traffic	4
4	Screenshots reflecting the progress	5
4.1	First Prototypes	5
4.2	More Advanced 3D Output	6
4.3	Added deferred rendering (first tries)	7
4.4	Deferred rendering and other shaders	8
4.5	Added more effects and sepia	9



2 Development state

2.1 Progress

Basically, we are (more or less) in schedule. We have successfully completed all low targets. At the moment, we are implementing desirable and high targets.

Even though we are basically following the development schedule of the formal game proposal, we have already implemented some of the high targets while not yet completed all desirable targets. Especially the graphics are already pretty advanced since we started to implement some shader effects.

3 Details on some targets

3.1 The Physics

After some research, we decided to use the Farseer Physics Library for our game's physics. Since this library is strictly for 2D physic calculations, it perfectly suits our need and does not have too much overhead coming from non-used features.

One problem we were facing was the implementation of a good driving experience with the "constraints" of Farseer. This process is not yet finished and some fine tuning of the car handling surely has to be done. Nevertheless, it already feels like a car and has the desired "pickup and play" characteristics.

All calculations of the physics are done with a fixed timestep inside a separate thread. This has mainly two reasons: The "main game thread" is discharged and we omit the occurrence of some overflows inside Farseer when doing larger delta-steps for fast moving objects.

3.2 Instancing

Our game features a lot of the same objects (mainly buildings and cars) on screen. For this reason, we decided to implement GPU instancing. This reduces the cost of repeated draw calls heavily. In combination with a good culling of non-used / non-drawn objects, we were able to reduce the CPU load.



3.3 Deferred rendering

In order to support multiple light sources we wanted to include deferred rendering. For example to render the headlights of the cars or the blinking lights of the police. To achieve this we followed a tutorial, which worked, apart from some problems with the z-buffer, nicely in our game.

For the headlights we had to extend the deferred render with spotlights and that's where the troubles started. In our first implementation nearly everything seemed to be wrong. That's where we discovered that shader debugging is really a pain in the ass. It turned out that the transformation of the cone, which has to be rendered for the spotlight, has been wrong. Therefore after fixing the transformation together with some minor bugfixes in the shader itself, the spotlights finally worked.

Some other graphical elements (marking of pickup and delivery points, the guidance system) are also implemented in shaders.

The "some how sepia effect" (we like to call it like that because it's not strictly a sepia – it still leaves more color on screen than a regular sepia filter would do) is rendered in the shader to.

3.4 AI for cops and civilian traffic

Since our game focuses on the single player experience, an (at least rudimentary) AI for the police cars is consequently a crucial part. The routes of the cops are calculated using an A* algorithm. We also have three different cop AI behaviour patterns:

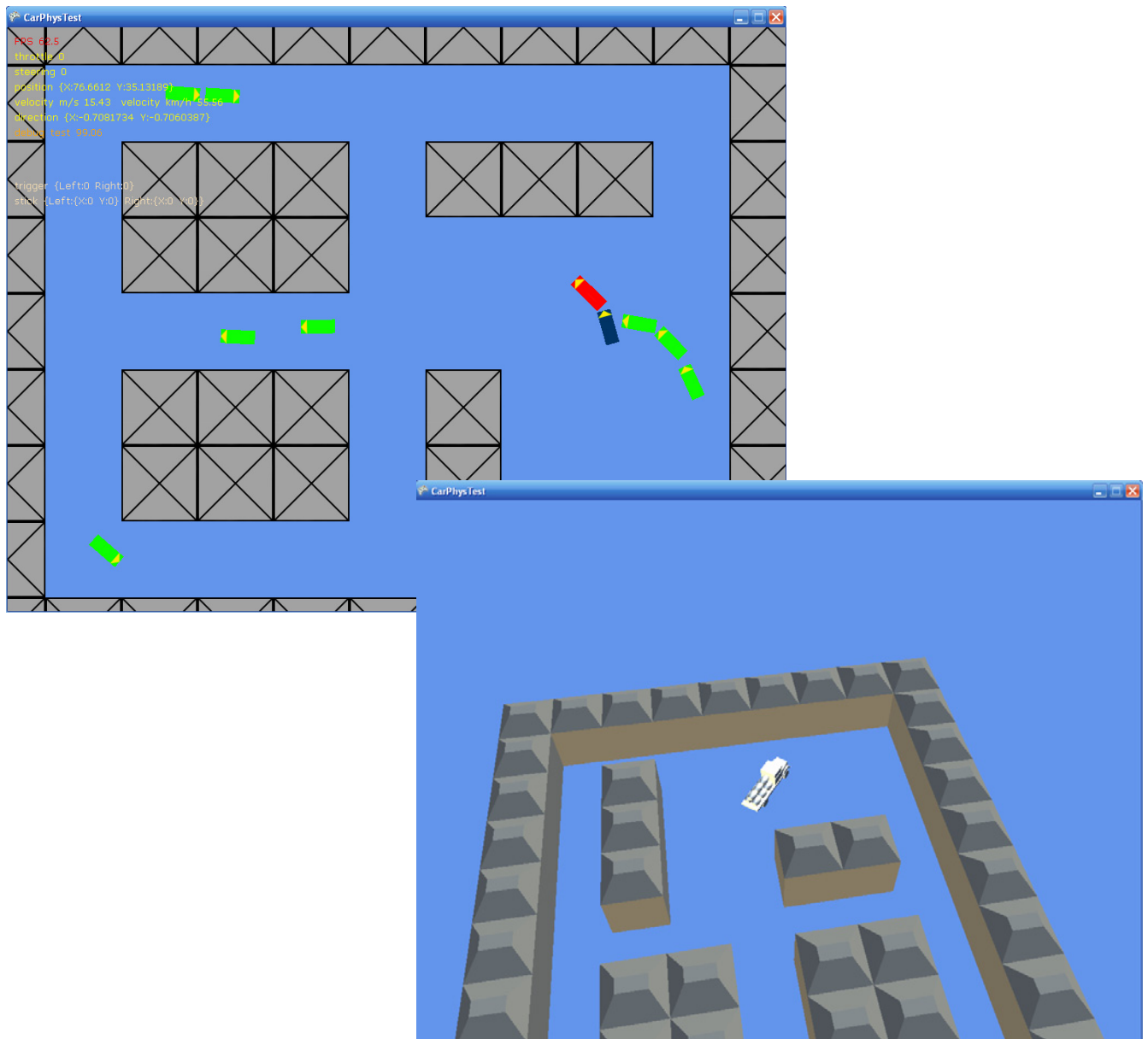
- Basic police AI cars just try to drive their car into the player's truck.
- Side police AI cars try to push the truck from the sides so that it's more difficult to steer.
- Frontal police AI cars try to overtake and brake in front of the truck.

The civilian traffic picks itself random points on the map as destinations and also uses the A* algorithm to calculate the corresponding route.



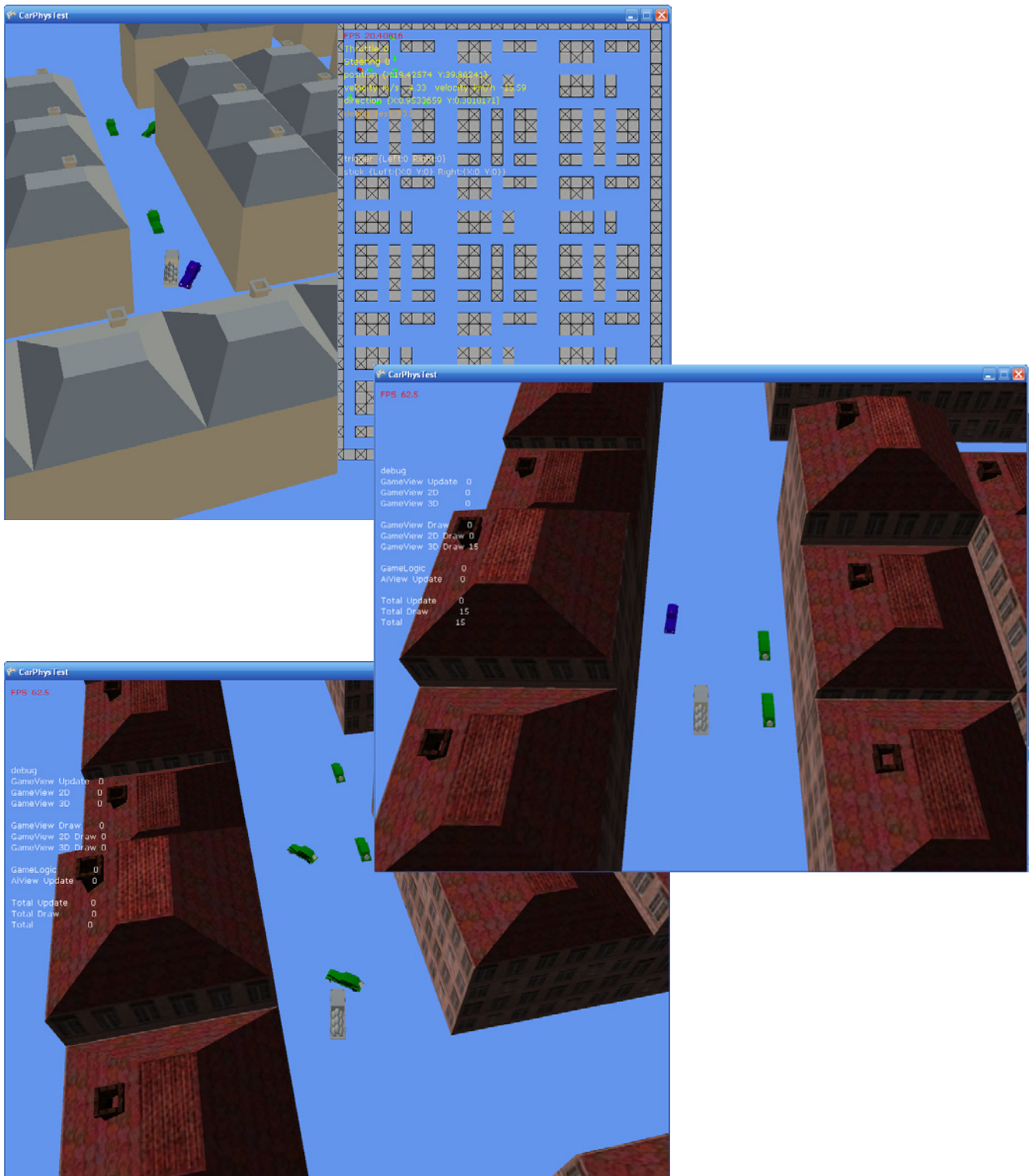
4 Screenshots reflecting the progress

4.1 First Prototypes





4.2 More Advanced 3D Output





4.3 Added deferred rendering (first tries)





4.4 Deferred rendering and other shaders





4.5 Added more effects and sepia

