Prof. Dr.-Ing. Ralf Steinmetz
Multimedia communications Lab

Dr. Florian Mehm
Dipl. Inf. Robert Konrad

TECHNISCHE
UNIVERSITÄT
DARMSTADT

„Game Technology“
Winter Semester 2014/2015

Exercise 2

For bonus points upload your solutions until **Friday the 31st of October 2014, 11:40**

General Information

- The exercises may be solved by teams of up to three people.
- The solutions have to be uploaded to the Git repositories assigned to the individual teams.
- **The submission date (for practical and theoretical tasks) is noted on top of each exercise sheet.**
- If you have questions about the exercises write a mail to game-technology@kom.tu-darmstadt.de or use the forum at https://www.fachschaft.informatik.tu-darmstadt.de/forum/viewforum.php?f=557

1. Practical Tasks: Crack me up (5 Points)

For reasons far beyond human comprehension old games were often accompanied by little start animations called cracktros. Most cracktros are simply functional animations of positions and colors. YouTube contains hundreds of examples.

Give your game the distinct TU Darmstadt style by adding a cracktro. Animate at least five properties of your graphics in a purely functional way.

https://github.com/KTXSoftware/Exercise2.git contains additional code to help you out. You can either copy the code changes manually or just pull them into your own repository using
git pull https://github.com/KTXSoftware/Exercise2.git

2. Theoretical Tasks: Timing (5 Points)

2.1 Iterative and functional time calculations

A space ship labeled Vic Viper starts at time 0 and x position 10 with an x speed of 14 per second. Calculate the x position after 2 seconds (a) functionally and (b) iteratively with 3 steps per second.

2.2 Different framerates

The situation starts out as described in 2.1 but there is a wall of width 5 at x position 60 (the space ship is a point, it has no width). Calculate the position of the space ship after 5 seconds for (a) a 3 steps per second and (b) a 2 steps per second update. Check for collisions in every update step. When a collision happens, move back the ship just so far that it barely touches the wall and set its speed to 0.

2.3 Framerate optimizations

You are working on a game without using any form of preemptive multitasking. As it turns out, a purely graphical effect you implemented runs a little slow. A colleague suggests to calculate and cache the effect every third frame only. Is that a proper optimization strategy? Discuss the resulting changes to the framerate.