Prof. Dr. Leif Kobbelt
Kersten Schuster, Philip Trettner

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Real-time Graphics Assignment 3

Date Published: November 15th 2017, Date Due: November 22nd 2017, 11:44

- The assignments have to be done in groups of 2 students.
- Hand in the solutions to the exercises via L^{2}P.
- You are only allowed to change code inside the marked strips (STUDENT CODE BEGIN/END)!
- Any questions? → L^{2}P discussion forum or rtg@cs.rwth-aachen.de!

If not done yet, obtain the (publicly accessible) exercise framework and assignments from `https://www.graphics.rwth-aachen.de:9000/Teaching/rtg-ws17-assignments/`.
Use **git pull** to fetch the newest changes of the framework (including the code for this exercise).

The **only** files that you should modify and **upload**:

- Assignment03.cc
- Tasks.cc

Each subtask corresponds to a code strip with more detailed comments and hints. Also consider the blackboard photos from the current exercise (L^2P).

Exercise 1 Extending the Pong rules [1 + 1 = 2 Points]

(a) We refine our equations of motion by adding acceleration and linear drag. For all `TransformComponent`s, compute the velocity from the acceleration. Before that, apply the appropriate linear drag to the acceleration (`updateMotionSystem(...)`). For more information, have a look at the `TransformComponent` struct in `Components.hh`.

(b) In `checkSphereSegmentCollision(...)`, the reflection direction should depend on where exactly the ball hits the paddle. For that, you can access a variable $t \in [0..1]$ that defines the local y-coordinate of the collision point on the paddle. Calculate a guiding vector $g := (n_x, 2t-1)^T$. The new velocity v' should have the same length as the incoming velocity and lie halfway between g and the reflected velocity.

Exercise 2 Developing an AI [2 + 2 + 2 = 6 Points]

In this exercise you will develop a computer player ("AI").

Note:

- If the ball moves away from your paddle we recommend moving to the center.
- In this exercise, linear drag and ball acceleration is disabled, there is only one ball at a time and there is only one paddle.

(a) In `ai::task2a(...)`, the AI-paddle (one the left) has to react to a ball that is spawned on the right side of the field and moves in horizontal direction. The only (non-local) value you can change is the return value of this method, the acceleration that is applied on the paddle.

- The ball should collide with the paddle center. You can check that by having a look at the values printed to the console: Everything between 45% and 55% should be fine (50% would be exactly the paddle center).

- We recommend looking up "Motion with Constant Acceleration" or "Gleichmäßig beschleunigte Bewegung" and solve this analytically.

(b) In `ai::task2b(...)`, the ball from the last task moves in a random direction, so that all reflections have to be taken into account. The restrictions from the former task still apply. This task is about predicting the ball movement.

(c) In `ai::task2c(...)`, the ball does not necessarily have to hit the paddle center, but rather should target exactly the center of the right side of the field. You can do this by exploiting the new mechanic from Exercise 1 (b).

Exercise 3 AI Battle [1 + 1 (+2) = 2 (+2) Points]

In this task you will use the AI pieces from exercise 2 and try to beat existing AIs. We consider an enemy AI (simple, normal, good) as beaten if it is clearly inferior, i.e. loses most of the time.

(a) Beat the simple AI.

(b) Beat the normal AI.

(c) Beat the good AI (Bonus).

Note: the good AI will be released (and announced) before the weekend.