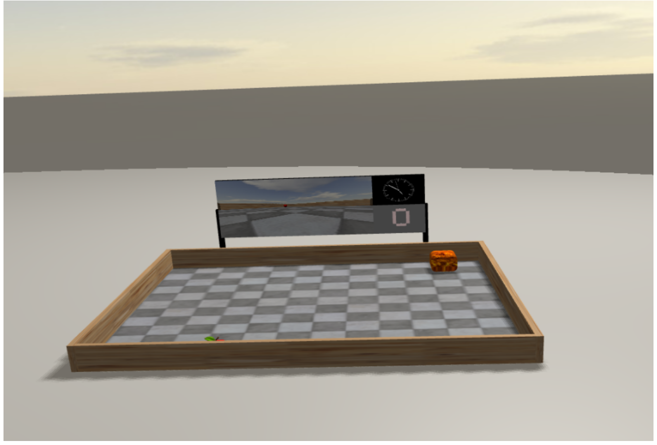
Introduction to Computer Graphics
K. Jünemann

Assignment 4: Spice up your snake game

In this lab you'll extend your snake game from lab 1 to make it a bit more interesting.



1 Requirements

1. Place the playing field on a large gray ground inside a sky box. Use the textures in the `skybox` subdirectory of the `resources` directory for the sky box.

2. Add a spot light and ambient light to the scene.

3. Increase your playing field to a size of 12 × 12 while leaving the size of a unit cell at 1 × 1. Also remove the camera motion. Position the main camera at a convenient initial position.

4. Apply the file `FloorsCheckerboard_S_Diffuse.jpg` as a texture map and the file `FloorsCheckerboard_S_Normal.jpg` as a normal map to the playing field. Configure the textures such that one unit cell of the playing field is covered by exactly one square of the checkerboard pattern.

5. Add walls of height 1 as boundaries to the playing field and apply the texture `hardwood2_diffuse.jpg` to the walls. Set the repeat-wrapping factor along the boundaries to a value of 3. Use the file `hardwood2_bump.jpg` as a bump map and set the bump scale to a value of 0.1.

6. Use cubes with rounded corners for the snake (see code example in the documentation of `THREE.ExtrudeGeometry`, use a value of at least 8 for the `bevelSegment` parameter). Apply the `lavatile.jpg` texture to the cubes. Use any colors you like for the snake cubes but make sure that the head is distinguishable from the rest of the snake.

7. Replace the red ball by the apple geometry defined in the file `Apple.obj`. Scale and position the apple object such that it is located on the playing field and conveniently fits into one unit cell. Apply the texture file `Apple_BaseColor.png` to the apple object. Also apply the file `Apple_Normal.png` as a normal map and the file `Apple_Roughness.png` as specular map.

8. Add a display board showing the snake game with a camera located at the head of the snake and looking into the direction of motion of the snake.

9. Show a small clock on the upper right corner of the display board. Feel free to use your own code from lab 2 for this.

10. Make sure that the snake, the apple, the boundary walls and the display board cast shadows on the ground and the playing field.

Optional: Add a counter to the lower right corner of the display board showing the current length of the snake.

2 Hints

- All textures files, the sound files and the Apple geometry definition file are located in the `resources` directory.

- If you find more suitable texture files, feel free to use them.

- For the optional counter you may use any font. The font file `kenpixel.ttf` which has been used for the video is provided in the `resources` directory. For this font type you also need the JavaScript files `TTFLoader.js` and `opentype.min.js` which are provided in the lib directory (see this example: https://threejs.org/examples/?q=TTF#webgl_loader_ttf).

3 Coding style

- Stick to the coding style guide which can be found in the `README.md` file for chapter 3 in the gitlab repository.
- It is your choice whether you load the `three.js` library as a module or just as a JavaScript text file.

4 Handing in the solution

No group work allowed. Every course partiticipant has to write her or his *own* code!
Implement your entire solution within the file `SnakeGame.js`. Feel free to change this file as you like but do not add any further files.

The deadline for submission to your git repository is the **8th of February**.