COMP 371 – Computer Graphics

Assignment 2: Particle Systems

Due: Friday, July 19th 2019, End of Day
Worth: 10% of your final grade

Topics

- Non-Programming: Spherical Coordinates and Viewport Transform
- Programming: Aligning Billboards with camera, parenting objects, interpolating particle system parameters over time and randomizing velocity within an angle in a cone

Programming Assignment [8 points]

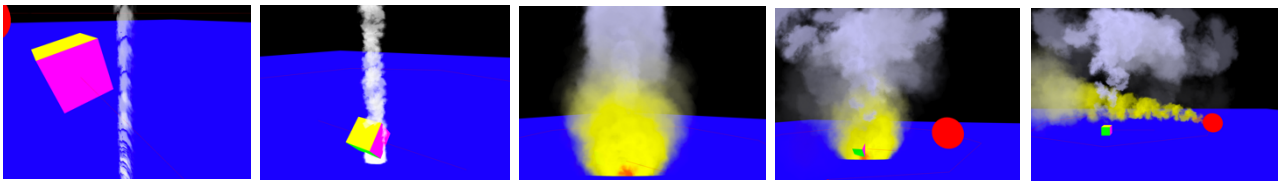

Figure 1 - Expected results after implementing required tasks

The programming assignment requires implementing missing functionalities in a Particle System. Originally, the particles are drawn on static flat plane (first image). To make them more realistic, you are tasked to render them as billboards, dynamically aligned towards facing the camera (second image). Then, particle attributes (acceleration, size and color) must be updated over time according to the particle descriptor (third image). Finally, you must implement a random velocity angle variation (fourth image) and parent particle emitters to models in the scene (fifth image).

Provided Files

Many files are provided with Assignment 2. Copy/overwrite these files in your Assignment 1 framework. When you open your Visual Studio or Xcode solution, make sure you add the new files provided in the project. A new scene is also provided that will create 2 particle emitters with different parameters.

Tasks to Do: (Look for @TODO in the code)

- Align billboards with Camera plane **[2 point]**
  - A billboard includes a position (center), a color, a size, an angle and a color.
  - A billboard renders with two triangles, 6 vertices in total, always facing the camera.
  - You can extract the camera basis vectors from the view matrix, rotate these vectors by the angle of the billboard along $\overrightarrow{lookAt}$, offset vertices from center by half the size of the billboard along the rotated right and up vectors.
  - Billboards also include a normal that will later be used for lighting. It must be facing towards the camera.

- Update particle parameters over time **[2 points]**
    - The velocity must be updated according to the particle acceleration.
    - The size of the particle must be updated with the size growth velocity in the particle descriptor.
    - The color must be updated over the particle lifetime according to three phases:
        1. (time < fade-in time): Color linearly interpolated from initial color to mid color.
        2. (fade-in time < time < lifetime - fade-out time): Color is mid color
        3. (time > lifetime – fade-out time): Color is linearly interpolated from mid color to end color.

- Randomize initial Particle Velocity vector **[2 points]**
    - The initial velocity vector will be randomly rotated by a random angle within a cone.
    - A good way to do this is to rotate along an axis perpendicular to the velocity vector by a random angle between 0 degree and the velocity angle randomness attribute.
    - Then, to make the rotation within a cone, you may do another random rotation between 0 and 360 degrees along the original velocity vector.

- Calculate particle emitter positions with parenting support **[2 points]**
    - A particle emitter may, or may not, be attached (parented) to another model.
    - If the emitter is parented, you must insure if follows its parent. The parent position and local position must both be considered.

Non-Programming Assignment [2 points]

(You can do it by hand with readable hand writing, and scan it with your assignment, or you can do it with a Word processor and submit it in PDF or DOCX format).

- From first principles, derive the conversion from Cartesian to Spherical Coordinates and from Spherical to Cartesian Coordinates. Your solution must include a clear picture of the triangles showing the mapping between the 2 coordinates system as seen in class. **[1 point]**
- From first principles, derive the viewport transform matrix by solving a system of linear equations mapping the NDC to Viewport planes as seen in class. **[1 point]**

Submission Guidelines

- You need to submit your assignment on Moodle by the assignment deadline.
- For the non-programming assignment, you will need to scan your readable hand written derivations, or include the PDF or DOCX word processed file.
- For the programming assignment, you should delete the Bin and Build folder before submitting your completed assignment.
- Everything must be submitted within a single ZIP file on Moodle.
- Anything missing from the Submission will result in grade deductions.