CGRA350 Real-time 3D Computer Graphics T2/2020

Assignment #3: Animation (20 points)
Assigned: 1st September 2020
Due: 21st September 2020

The main objective of this assignment is to create a system for key frame animation of rigid objects. You will need to create key frames to define movements of the objects and generate in-between motion using proper interpolation. The goals of this assignment are:

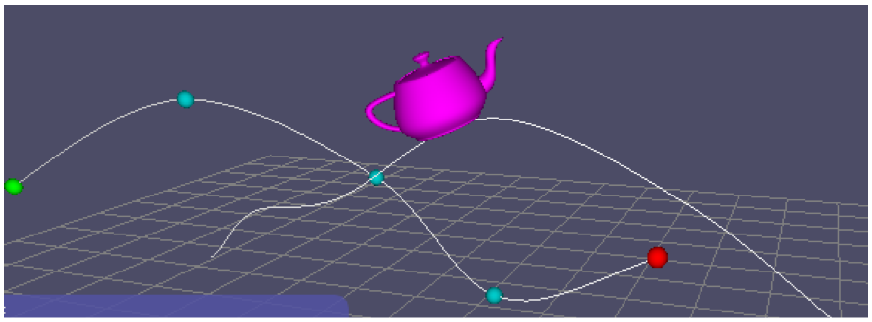

Figure 1: An example of an object animation system.

- Implement a key frame animation system to animate a pure translation motion
- Understand arc-length parametrization and implement speed control for keyframe animation
- Find the derivatives of a spline and use them to orientate objects along a path.

Turn in procedure

You should submit your work as a zip file using the ECS submission system. Please name your file as `<LastName><FirstName>-Assignment<X>.zip` where X is the assignment number. When your file is unzipped you should have:

1. The C++ and shader programs you have written. You should use files in the form of the samples given, rather than producing files from scratch. This will help us follow your code.
2. Your report in PDF format (either typed directly or scanned in).
3. If the question asks you to write code to make images, provide sample images created by your program. You can save these by taking a screenshot. Alternatively you can include these in your PDF report.

4. Any other information or supporting documentation to help us run and evaluate your submission.

If your programs fail on the machines used for grading, you may be asked to bring in your system to demonstrate that the files you submitted functioned in the environment you worked in.

Part 1: Key-frame animation (18 points total)

Core (11 points)

1. Create and draw a Catmull-Rom spline in world-space. Define a spline with at least 5 unique control points. Draw a sphere at each control point, with the start and end control points colored green and red respectively. To draw the path of the spline, create a mesh with the `GL_LINES` enum, and fill the mesh with vertices that have been interpolated part-way along the spline.

2. Animate an object's position and rotation using a Catmull-Rom spline. Each control-point in the Catmull-Rom spline should define a position and the object should be rendered by using interpolated points as its position in world-space.

3. Animate the camera position using a Catmull-Rom spline for the position and have it rotated towards the object while animating. This means that while both the object and the camera are moving separately, the camera should "track" the objects movements wherever it is. Make sure that your camera is always still oriented vertically (that is, "up" does not change).

Completion (5 points)

Without special consideration, the speed of animation of each segment of a Catmull-Rom spline is equal. This means the object will take the same amount of time traversing a short distance as it does over a long distance, giving the impression that it changes speed (distance over time).

1. Reparameterize the spline to animate the object and camera at a constant speed. This can be done using arc-length reparametrization of the curve.

2. Illustrate your implementation by animating two objects simultaneously, where one uses the original curve parametrization and the other uses the new one.

3. Explain your algorithm in the report.

Challenge (2 points)

Add rotation to the object following the spline so the front of the object points in the direction of the path it is following along the spline. Just like the camera, ensure that the object always stays upright.

This will require you to use an object whose orientation is obviously identified. We recommend using the `teapot.obj` that we have supplied.

Part 2: Writing (2 points total)

Report (2 points)

Submit a 1-2-page PDF document outlining your experiences. In this report, you should include:

- Brief introduction of your functions in your programs, including how to run your program to get the desired results.
- The results of core, completion and challenge (depending on how much you have done) including images produced by your programs.