Assignment Three
Due Date: 17 March 2018

February 2018

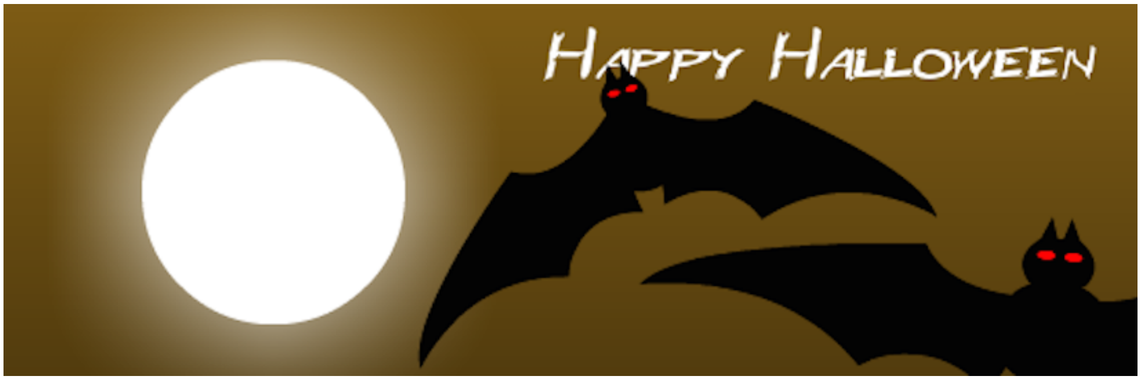

Figure 1: Screenshot of example animation

1 Introduction

In this assignment you will create a simple OpenGL 2D application to create a flash animation like the one shown in here. The animation does not have to necessarily be what is in the video... be creative!

1.1 Basic Functionality (80%)

The basics can be separated into two different executables (to simplify your life)

- the first uses simple geometry and transformations to create a small animation (start with triangles, later use Bezier)
  - one level of transformation hierarchy (e.g. flapping wings)
  - one animation bezier path (e.g. translation/rotation/scale of bat around scene)
- the second is for editing Bezier curves (number of vertices and initial layout can be hard-coded)
  - editor that allows to change Bezier handle positions and tangents
- (drawing text is not required, but you can just load simple texture for it if you want)

1.2 Advanced Functionality

Note many operations can either be performed on the CPU or the GPU (CPU/GPU points for the same feature do not cumulate).

Modeling

- generate content with convolutions (see sun halo in teaser image) (1% CPU, 2% separable filter, 3% GPU, 4% separable filters on GPU)
- rasterize the bezier polygon into a polygon (render the bezier boundary to a framebuffer, then use parity check to perform inside/outside on each scanline) (5%)
- triangulate a bezier polygon with trapezoidal decomposition (5%)
- use textures with alpha-blending (e.g. the wing is just a textured quad where some of the pixels are just transparent) (2%)
- efficient Bezier evaluation (recursive deCastlejau split rule) (2%)
- use tessellation shaders (OpenGL4 only) to draw bezier curves efficiently (5%)

Effects

- anti-alias each frame of your animation (clearly you cannot use GL_MULTISAMPLE) (3% CPU, 5% GPU)
- motion blur moving objects (not just global motion blur!) (2%)

Animation

- arc-length parameterization of motion trajectories (2%)
- easy in/out for the animation path (1%)
- save your animation as a MOV/AVI
  - just use some screen capture software (1%)
  - dump framebuffer as images then load the image sequence as video (2%)
- let it snow!! particles, geometry shaders and sprites (5%)
- build a system that allows animation key-framing editing (10%)

Software Engineering

- code quality (e.g. organization, Object Oriented programming) (1%)
- use of git repository (i.e. frequent commits with quality log messages) (1%)
- software documentation (comments) (1%)

2 Getting Started

Suggestion: start the assignment by animating a few simple triangles (e.g. the wing of the bat can be approximated by triangles). Later you can replace these with Bezier curves, polygons and textures.