Interactive Evolutionary Character Animation

Interactive evolutionary design and ideation tools are especially useful when the user is not quite sure what style will work best and wants to explore options, or when the user wants to generate sets of digital assets that are unique but share stylistic qualities. I built the parametric walk cycle and the interactive evolutionary ideation tool in Maya (python and mel) to help crowd animators generate custom sets of movement ideas. (publication)

Trace Selection

This animation tool uses subsequence dynamic time warping to match a user's trace gesture with the most similar part of an animated trajectory in the scene. Once part of an animation is selected, the trace can be translated to edit the spatial characteristics of the movement. I created this tool in Maya and also in Houdini (with python). For further details or more examples, see the full video and the publication.

Targeted Smoke

User-defined images provide the keyframe targets for the smoke optimization. This effects tool is an implementation of the technique of Fattal & Lischinski (SIGGRAPH 2004) and was built on top of Jos Stam's 2D fluid solver demo code. (OpenGL and C/C++)

Interactive Evolutionary Prop Design and Paint Layout

This interactive evolutionary prop design tool enables layout artists to explore parametric design spaces. I created a parametric cactus digital asset in Houdini to demonstrate the user experience. At each iteration, cactus prop designs are sorted using multidimensional scaling, and sensitivity analysis enables region selection. (publication) The layout tool allows for paint-style interaction and generates props that inherit the settings of an evolved set of prop designs from the evolutionary design tool. The parameter settings can be swapped out after painting for comparison if desired.

Spot Light Attribute Shaping Tool

This Maya scripted plugin (python) allows the user to edit light attribute keyframes directly in the 3D scene instead of having to edit them in the graph editor where the user cannot see how the light interacts with other objects in the scene.

Ray Tracer

Starting with only a scene graph library, I wrote a ray tracer which calculates illumination, generates Perlin noise textures, and computes soft shadows. (OpenGL and C++)

Hand Drawn Work

As a programmer of computer graphics tools I enjoy experiencing a range of hands-on methods for making art and how they can inform approaches to human interaction with new applications.

[Link to a portfolio of other work done in collaboration with educators, choreographers, and designers]