An Online Collaborative Ecosystem for Educational Computer Graphics

Garett Ridge, Demetri Terzopoulos
UCLA
“Demopedia”
Prototype:
https://web3d-2019.glitch.me/
Similar Websites

* sites that do not have in-browser coding

GitHub
CODEPEN
Instant Hosting
Demopedia
Glitch
Shadertoy

Graphical

babylon.js
p5.js
Code.org

Educational

Wikipedia
The Free Encyclopedia
Technical skills required

- JavaScript
- tiny-graphics.js file
What’s tiny-graphics.js?

- Made using feedback from around 1000 students
- A small single file that defines wrappers around WebGL

GitHub:

https://github.com/encyclopedia-of-code/tiny-graphics-js
What’s tiny-graphics.js?

- Graphics code divides up naturally into OOP units
Using tiny-graphics.js

```javascript
model_transform = Mat4.identity();

this.shapes.box.draw( context, program_state, model_transform, this.materials.plastic );

model_transform = model_transform.times( Mat4.rotation ( 1, 0,0,1 ) )
    .times( Mat4.scale ( 1, 2, 1 ) )
    .times( Mat4.translation( 0,-1.5, 0 ) );

this.shapes.ball.draw( context, program_state, model_transform, this.materials.plastic );
```
tiny-graphics.js UI widgets

“Canvas widget”
This demo shows nodes of a falling 1D string. It steps a simulation of Hooke's law in fixed time increments using the forward-Euler method. Use the buttons below to create the illusion to the simulator that the display framerate is running fast or slow, independent of the simulation time step size. The simulation's answers will be unaffected.

Parts of this demo are left for you to fill in as an exercise. To improve the simulation, follow the instructions in the "TODO" code comments, adding in a small non-zero rest-length and a series damper to each spring segment.

Right now the first node is constrained to stay in place, due to zeroing its velocity every timestep. Near the last "TODO" comment, try fixing more than one node in place. Also observe what happens if you fix none.

Reduce the fixed timestep in the code (called dt) to 1/5000 seconds, causing more discrete physics steps to be computed. What happens to the frame rate? Does the spring's movement speed change accordingly or does it mostly stay the same?

Lastly, change the timestep back and see how many nodes you can add to the string before the forward-Euler method breaks down and numerical error of your added damping term catastrophically accumulates, exploding the simulation vector to infinity.
tiny-graphics.js UI widgets

“Controls widget”
tiny-graphics.js UI widgets

“Code widget”

```javascript
const blue = color( 0,0,1,1 ), yellow = color( 1,1,0,1 );

// Variable model_transform will be a local matrix value that helps us position shapes.
// It starts over as the identity every single frame - coordinate axes at the origin.
let model_transform = Mat4.identity();

// Draw a hierarchy of objects that appear connected together. The first shape
```
tiny-graphics.js UI widgets

“Editor widget”

class My_Scene
{
}

tiny-graphics.js UI widgets

- Widgets are easily embedded into any page
- Widgets can be nested in each other
Code examples given to students so far

Modeling

Rendering

Animation

Interaction
Teaching modeling hierarchies and matrix order with the “Matrix Game”

http://bases-game.glitch.me/
Results

- With our resources, student projects reached farther topics in the graphics curriculum than before.
- The students said in anonymous surveys that our resources saved them time and labor.
- New modes of graphics programming online.
Sample of student projects
One student project exemplified how to teach with WebGL-enabled documents:

https://intro-graphics-master.github.io/term-project-8/

by:
Junhong Wang
John Tran
Jonathan Liau

Particle Effects

A particle is really just a 2D square in a 3D scene. To make it look like it’s in 3D scene, the particles have to be small enough, we can simply apply model transform such that it cancels out the camera rotation. To animate the particle, we keep the lifetime of the particle.
“Active Textbooks”

A page with a series of multiple WebGL panels, that incrementally tell a story.

Features:

● Documents that are able to run a program while you read them.
● Interact with rich 3D scenes in several document regions while you learn.
● Visualize facts on the web that can be conveyed better in 3D.
Seamless sharing of graphics objects between WebGL contexts and panels
Seamless sharing of graphics objects between WebGL contexts and panels
Seamless sharing of graphics objects between WebGL contexts and panels
Seamless sharing of graphics objects between WebGL contexts and panels
Sample “Active Textbook” link


Parametric surfaces can be wrapped around themselves in circles, if increasing one of $v$ or $u$ causes a rotation around an axis. These are called surfaces of revolution.

To draw these using Grid.Patch, we provide another class called Surface_Of_Revolution that extends Grid.Patch and takes a set of points as input. Surface_Of_Revolution automatically sweeps the given points around the Z axis to make each column. Your list of points, which become the rotes, could be arranged to make any 10 curve. The direction of your points matters; be careful to orient up with your normal vectors at pointing inside out after the sweep.
Future work - next goals

- Crowd-sourcing more graphics utility functions
- Delivering flashy games to draw people in
Future work - next goals

- Delivering users and clients better tutorial documents than they had before
  - Interactively tell a visual story with many WebGL panels

- Making there be people who "learned how to program" from our tutorials
Thanks!