Graphics Programming Education

Less Is More

John Pile Jr
Asst Professor of Game Programming
Game Studio | Champlain College

GDC EDUCATION SUMMIT

GAME DEVELOPERS CONFERENCE
SAN FRANCISCO, CA
MARCH 17–21, 2014
EXPO DATES: MARCH 19–21
2014

Reminders:

Cell phones

Questions

25 minutes

Wrap-up Room (West 3000)

John Pile Jr

Less is More:

Improve the success of undergraduate students learning graphics programming…

…by replacing a *30-week* **3D** graphics programming sequence with a *15-week* **2D** course, followed by a *15-week* **3D** course.

Topics

1. About our program
2. Challenges w/ trad. graphics courses
3. Curriculum changes
4. 2D graphics course/topics
5. Results (expected and unexpected)
6. Q/A

CHAMPLAIN COLLEGE

1878

4-year Undergraduate

Private | Non-Research

Game Art & Animation
Game Design
Game Programming
Mgmt of Creative Media

Previous Curriculum

Year 1

Computer Theory
Game History & Development
Calculus

C++ Programming I
Game Technology
Matrices, Vectors, & 3D Math

Year 2

C++ Programming II
Intro to Networking & Security
Discrete Math

Game Architecture
Game Production I
Data Structures

Year 3

Graphics I (3D)
Linux / Unix
Computer Systems for Soft Eng

Graphics II (3D)
Game Production II
Elective

Year 4

Game Capstone
AI for Games
Networking for Games

Game Physics
Senior Production
OS Architecture

+ an additional 48 credits of core liberal arts

Year 1

Computer Theory

Game History & Development

Calculus

C++ Programming I

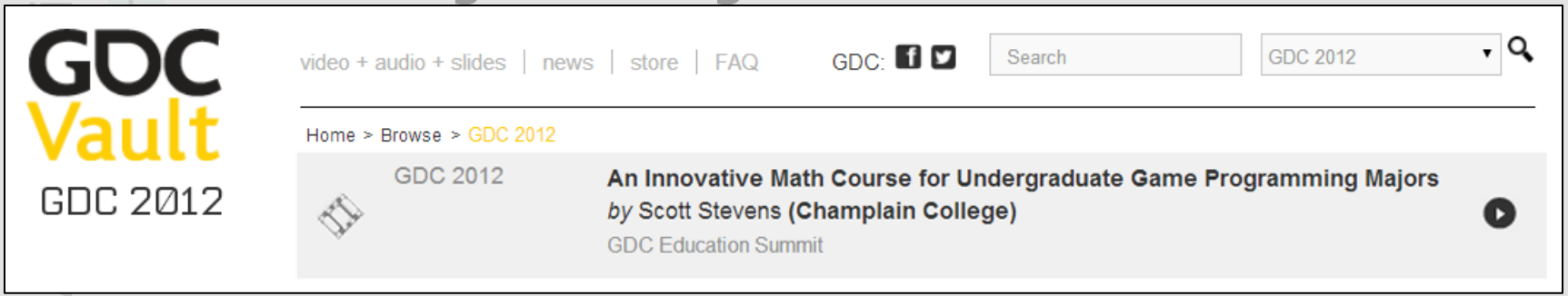
Game Technology

Matrices, Vectors & 3D Math

Year 2

C++ Programming II

Game Architecture



Year 4

Game Capstone

AI for Games

Networking for Games

Game Physics

Senior Production

OS Architecture

+ an additional 48 credits of core liberal arts

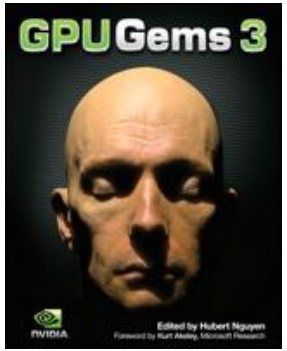
Previous Curriculum

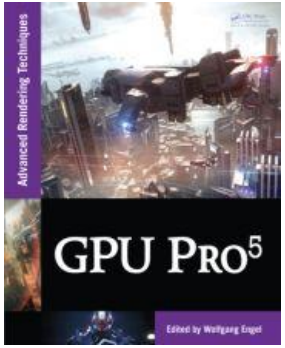
Year 1	Computer Theory	C++ Programming I
	Game History & Development	Game Technology
	Calculus	Matrices, Vectors, & 3D Math
Year 2	C++ Programming II	Game Architecture
	Intro to Networking & Security	Game Production I
	Discrete Math	Data Structures
Year 3	**Graphics I (3D)**	**Graphics II (3D)**
	Linux / Unix	Game Production II
	Computer Systems for Soft Eng	Elective
Year 4	Game Capstone	Game Physics
	AI for Games	Senior Production
	Networking for Games	OS Architecture

+ an additional 48 credits of core liberal arts

Goal: Undergraduate Graphics Programming Education

A solid foundations in fundamentals…



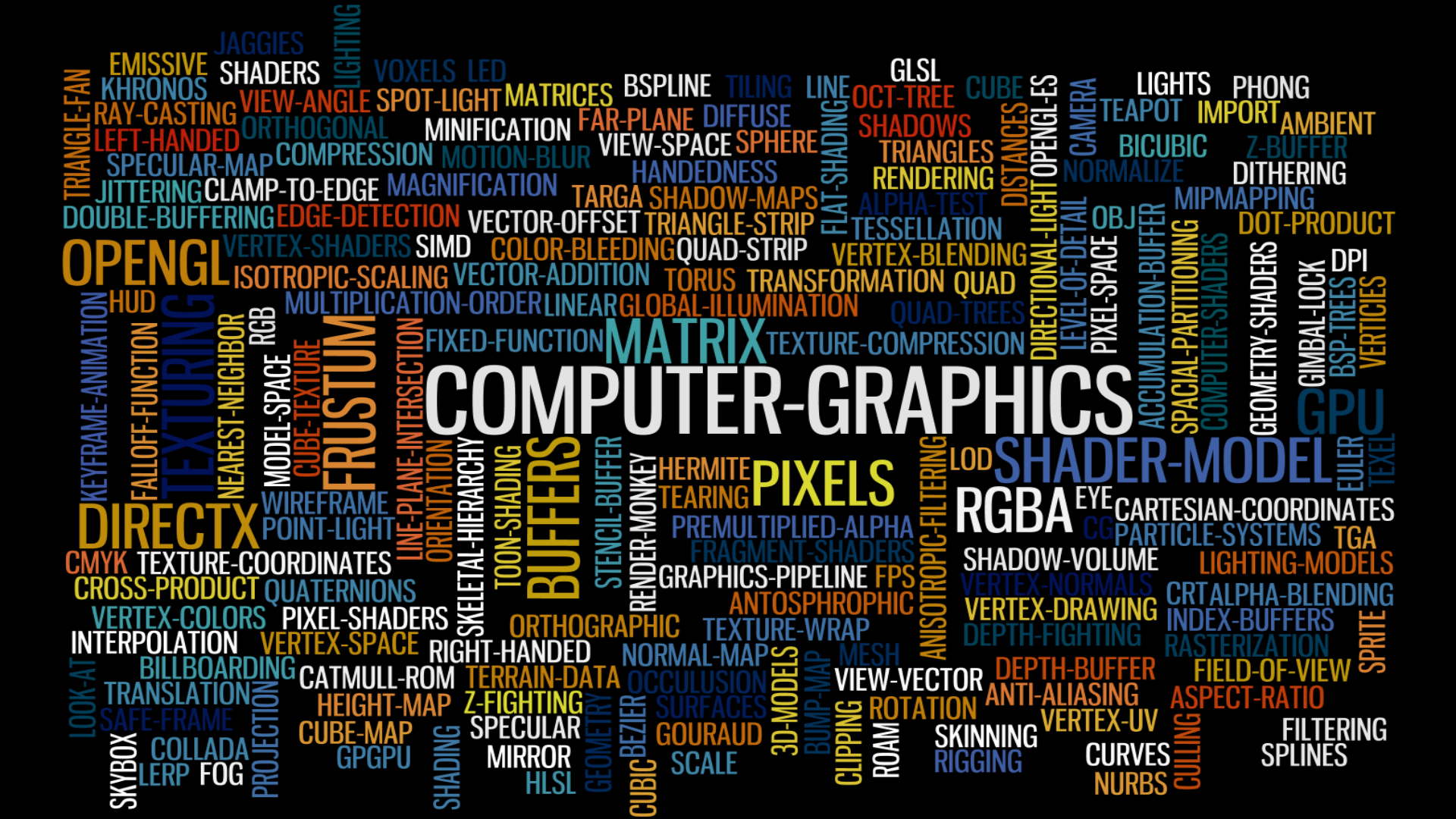


My Teaching Philosophy

self-driven, project-based learning
with a hint of competition

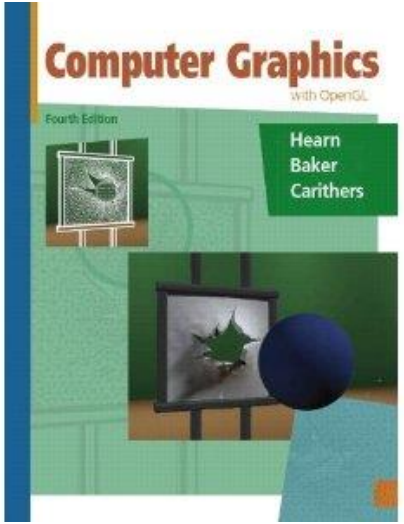
Graphics Programming Challenge

- Solid C++ skills
  - Data Structures, File IO, Use of APIs
- Good 3D Math Skills
  - Matrix and Vector Math
- Significant Programmer Confidence
  - When something goes wrong (scale, orientation, etc)
  - Ready for paradigm shift (GPU programming)
- Huge amount of “base knowledge”


COMPUTER-GRAPHICS
OPENGL
MATRIX
PIXELS
SHADER-MODEL
BUFFERS
TEXTURING
FRUSTUM
DIRECTX
RGBA
GPU
JAGGIES
EMISSIVE
SHADERS
VOXELS
LED
LIGHTING
KHRONOS
BSPLINE
TILING
LINE
GLSL
CUBE
LIGHTS
PHONG
TRIANGLE-FAN
RAY-CASTING
VIEW-ANGLE
SPOT-LIGHT
MATRICES
OCT-TREE
TEAPOT
IMPORT
AMBIENT
LEFT-HANDED
ORTHOGONAL
MINIFICATION
FAR-PLANE
DIFFUSE
SHADOWS
CAMERA
BICUBIC
Z-BUFFER
SPECULAR-MAP
COMPRESSION
MOTION-BLUR
VIEW-SPACE
SPHERE
TRIANGLES
DISTANCES
FLAT-SHADING
OPENGL-ES
HANDEDNESS
RENDERING
NORMALIZE
DITHERING
JITTERING
CLAMP-TO-EDGE
MAGNIFICATION
TARGA
SHADOW-MAPS
ALPHA-TEST
MIPMAPPING
DOUBLE-BUFFERING
EDGE-DETECTION
VECTOR-OFFSET
TRIANGLE-STRIP
TESSELLATION
OBJ
DOT-PRODUCT
VERTEX-SHADERS
SIMD
COLOR-BLEEDING
QUAD-STRIP
VERTEX-BLENDING
DIRECTIONAL-LIGHT
LEVEL-OF-DETAIL
PIXEL-SPACE
ACCUMULATION-BUFFER
ISOTROPIC-SCALING
VECTOR-ADDITION
TORUS
TRANSFORMATION
QUAD
COMPUTER-SHADERS
GEOMETRY-SHADERS
GIMBAL-LOCK
DPI
HUD
MULTIPLICATION-ORDER
LINEAR
GLOBAL-ILLUMINATION
QUAD-TREES
BSP-TREES
VERTICIES
KEYFRAME-ANIMATION
RGB
FIXED-FUNCTION
TEXTURE-COMPRESSION
FALLOFF-FUNCTION
NEAREST-NEIGHBOR
MODEL-SPACE
CUBE-TEXTURE
LINE-PLANE-INTERSECTION
ORIENTATION
TOON-SHADING
STENCIL-BUFFER
HERMITE
TEARING
RENDER-MONKEY
LOD
EULER
TEXEL
WIREFRAME
POINT-LIGHT
PREMULTIPLIED-ALPHA
EYE
CARTESIAN-COORDINATES
CMYK
TEXTURE-COORDINATES
FRAGMENT-SHADERS
CG
PARTICLE-SYSTEMS
TGA
CROSS-PRODUCT
QUATERNIONS
SKELETAL-HIERARCHY
GRAPHICS-PIPELINE
FPS
SHADOW-VOLUME
LIGHTING-MODELS
VERTEX-NORMALS
CRT
ALPHA-BLENDING
VERTEX-COLORS
PIXEL-SHADERS
ANTOSPHROPHIC
VERTEX-DRAWING
INDEX-BUFFERS
INTERPOLATION
VERTEX-SPACE
ORTHOGRAPHIC
TEXTURE-WRAP
DEPTH-FIGHTING
RASTERIZATION
RIGHT-HANDED
NORMAL-MAP
MESH
ANISOTROPIC-FILTERING
SPRITE
BILLBOARDING
CATMULL-ROM
TERRAIN-DATA
OCCULUSION
VIEW-VECTOR
DEPTH-BUFFER
FIELD-OF-VIEW
LOOK-AT
TRANSLATION
HEIGHT-MAP
Z-FIGHTING
SURFACES
ANTI-ALIASING
ASPECT-RATIO
SAFE-FRAME
CUBE-MAP
SPECULAR
GOURAUD
ROTATION
VERTEX-UV
FILTERING
SKYBOX
COLLADA
GPGPU
MIRROR
SCALE
SKINNING
CURVES
SPLINES
LERP
FOG
HLSL
RIGGING
NURBS
PROJECTION
SHADING
GEOMETRY
BEZIER
CUBIC
3D-MODELS
BUMP-MAP
CLIPPING
ROAM
CULLING

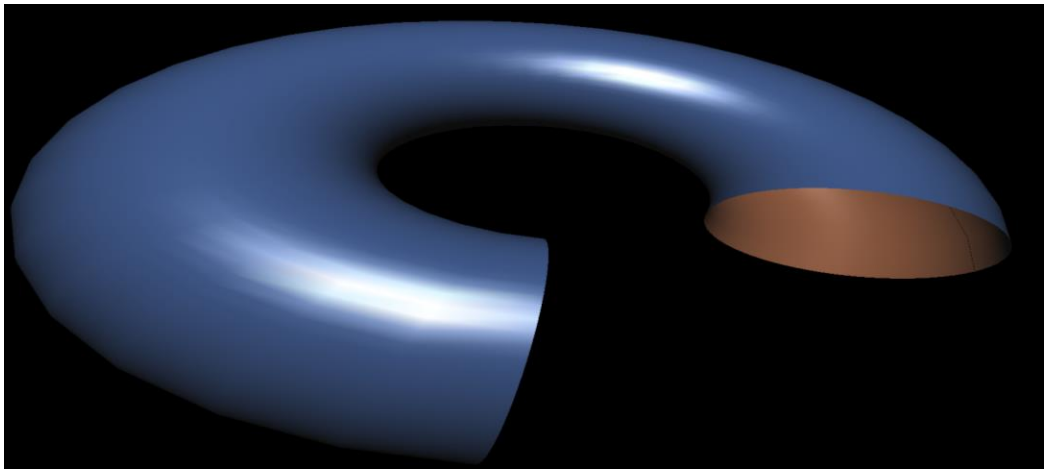
Traditional 3D Graphics Education

- Missing "Game Topics"
  - UI
    - Fonts & Localization
    - Menus, Transitions, & Safe-frames
  - Polish
  - Giving Control to Designers



Traditional 3D Graphics Education

- Time wasted on “traditional” CS topics
  - Spend time programmatically generating shapes



Traditional 3D Graphics Education

- Missing the "Value of Artist"
  - 3D models formats (not .OBJ)
  - Artists write their own shaders
  - Every system needs an editor
    - terrain
    - particle systems
    - fonts
    - tools integration


```
<COLLADA xmlns="http://www.collada.org/2005/11/COLLADASchema" version="1.4.1
 <asset>
   <contributor>
     <author>gcorson</author>
     <authoring_tool>Maya 8.0 | ColladaMaya v3.02 | FCollada v3.2</authoring_t
     <comments>Collada Maya Export Options: bakeTransforms=0;exportPolygonMesh
curveConstrainSampling=0;exportCameraAsLookat=0;
exportLights=1;exportCameras=1;exportJointsAndSkin=1;
exportAnimations=1;exportTriangles=0;exportInvisibleNodes=0;
exportNormals=1;exportTexCoords=1;exportVertexColors=1;exportTangents=0;
exportTexTangents=0;exportConstraints=1;exportPhysics=0;exportXRefs=1;
dereferenceXRefs=0;cameraXFov=0;cameraYFov=1</comments>
     <source_data>
file:///C:/Documents%20and%20Settings/gcorson/Desktop/Physics%20Tests/New_Proj
     </source_data>
   </contributor>
   <created>2007-04-04T23:08:51Z</created>
   <modified>2007-04-04T23:08:51Z</modified>
```


MAYA

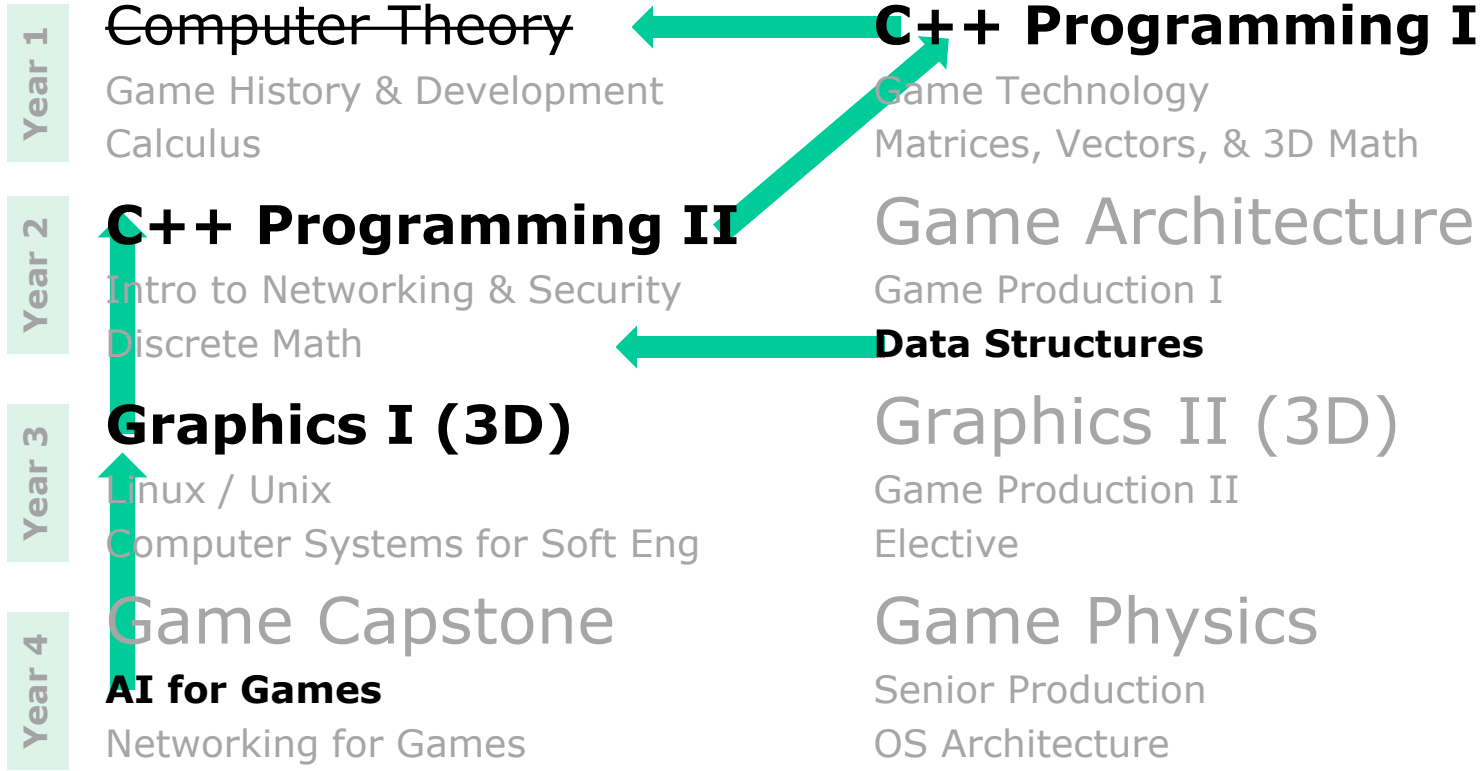
Summary:

Can’t be offered until 3rd year.

Even then, the average student is overwhelmed … then disillusioned.

No space for additional course.

Previous Curriculum



+ an additional 48 credits of core liberal arts

Updated Curriculum

Year 1

C++ Programming I

Game History & Development
Calculus

Year 2

Graphics I (2D)

Data Structures

Discrete Math

Year 3

AI for Games

Linux / Unix
Computer Systems for Soft Eng

Year 4

Game Capstone

Networking for Games
Elective

Stronger C++ Programmers

C++ Programming II

Game Technology
Matrices, Vectors, & 3D Math

Game Architecture

Game Production I
Intro to Networking & Security

Graphics II (3D)

Game Production II
Elective

Game Physics

Senior Production
OS Architecture

+ an additional 48 credits of core liberal arts

Updated Curriculum

Year 1

C++ Programming I
Game History & Development
Calculus

C++ Programming II
Game Technology
Matrices, Vectors, & 3D Math

Year 2

Graphics I (2D)
Data Structures
Discrete Math

Game Architecture
Game Production I
Intro to Networking & Security

Better AI Application in Production Courses

Year 3

AI for Games
Linux / Unix
Computer Systems for Soft Eng

Graphics II (3D)
Game Production II
Elective

Year 4

Game Capstone
Networking for Games
Elective

Game Physics
Senior Production
OS Architecture

+ an additional 48 credits of core liberal arts

Updated Curriculum

Year 1

C++ Programming I
Game History & Development
Calculus

C++ Programming II
Game Technology
Matrices, Vectors, & 3D Math

Year 2

Graphics I (2D)
Data Structures
Discrete Math

Game Architecture
Game Production I
Intro to Networking & Security

Year 3

AI for Games
Linux / Unix
Computer Systems for Soft Eng

Graphics II (3D)
Game Production II
Elective

Year 4

Game Capstone
Networking for Games
Elective

Game Physics
Senior Production
OS Architecture

Better 3D Graphics Programmers

+ an additional 48 credits of core liberal arts

Graphics I

Graphics Programming in 2D?

Language and API?

C# | XNA

Any language w/ any graphics library.

Graphics I (2D) – Game Topics

- Artist
  - Loading assets from file in specific formats (sprites) from a sprite editor
    * XML - Collada prep
  - Creating an editor for particles
- Loading tiled level
  * heightmap prep
- Polish
  - UI, Fonts, Transitions

Graphics I (2D) – 3D prep

- Introduction to the API (XNA, DirectX, OpenGL)
- Color, alpha-blending functions
- Display Buffer and double buffering
- Tracking depth and draw-order (intro to depth buffer)
- Image-file formats, image compression
- Filtering – magnification/minification techniques
- Review of Vector Math (addition, normals, dot products)
- Camera location, Scaling Matrix
- Optional: Rotation and Translation in 2D with Matrices

Graphics I (2D) – 3D prep

- Triggering Animation Cycles
- Building and performance of a particle system
- Curves – Interpolation, Splines
- Vector Graphics – Intro to rasterization (GPU)
- Pixel-perfect Collision Detection - Spacial Partitioning

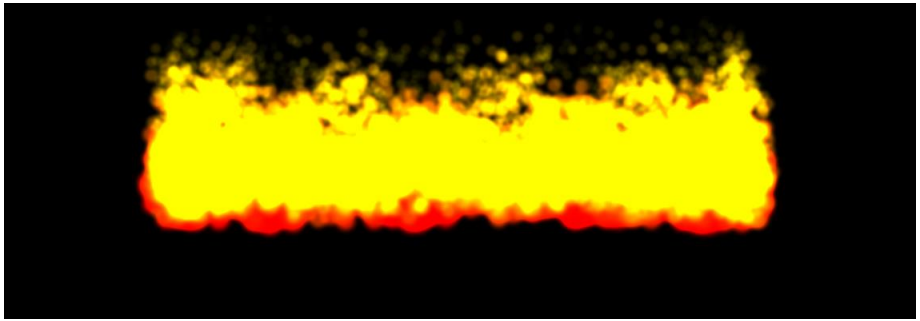
Graphics I (2D) – 3D prep

- GPU programming via pixel/fragment shader
  - Applications:
    - blur techniques
    - anti-aliasing
    - bloom effect
    - upscaling interpolation (linear, nearest neighbor, etc)
  - Topics:
    - attributes (passing data from CPU to GPU)
    - accumulation buffers and masking
    - texture sampling
    - performance monitoring

Project-based Learning

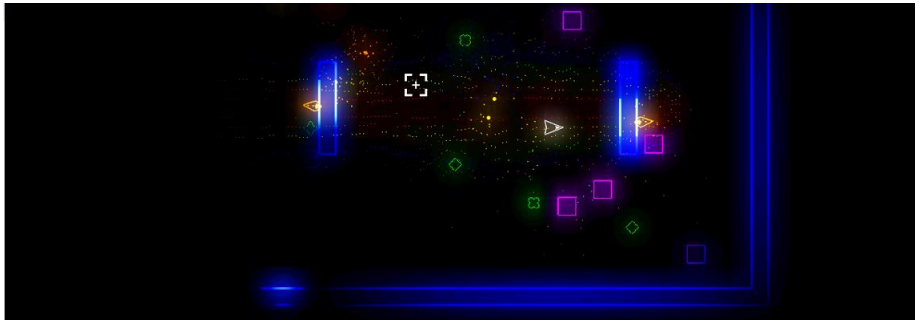
- Various Projects
  - Minor (1 week)
    - sprite animation, importers, level editors, etc
  - Major (2-4 weeks)
    - Particle System
    - Pixel Shader

Major Projects



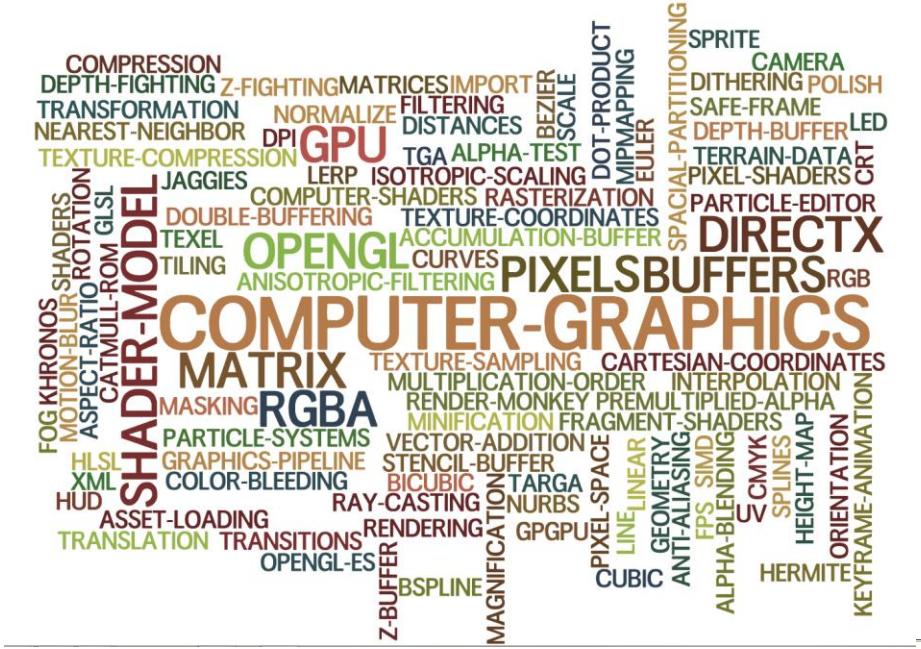
- Particle System
  - multiple effects
  - include an editor
  - deploy to low-end hardware
  - competitive elements:
    - performance
    - visuals
    - interaction

Major Projects



- Pixel Shader
  - multiple effects
  - deploy to low-end hardware
  - competitive elements:
    - visuals
  - research element

2D Course Topics



3D Course Topics

Outcomes?

Outcomes

- Improved average student performance
- Improved quality of 3D applications
  - Higher level of polish
  - Better understanding of “tech demos”
- We now have the numbers required to offer an “advanced graphics” course

Outcomes

- Opportunities for success for others
  - tools development & integration
  - low-end API work

Outcomes

- Improved Retention
  - students see relevancy sooner

    performance ->

    data structures ->

    computer architecture

Unexpected Outcomes

- Game Architecture
  - not distracted by graphics programming

Unexpected Outcomes

- Game Design Students

Employment Outcomes

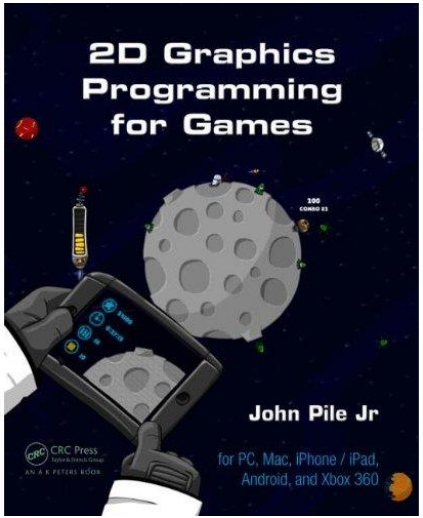
With new curriculum: (2011 graduated in 2013)

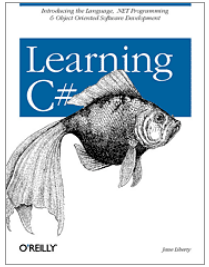
100% Employment including:

Graphics Software Engineer	WB/Turbine
Rendering Software Engineer	Apple Inc

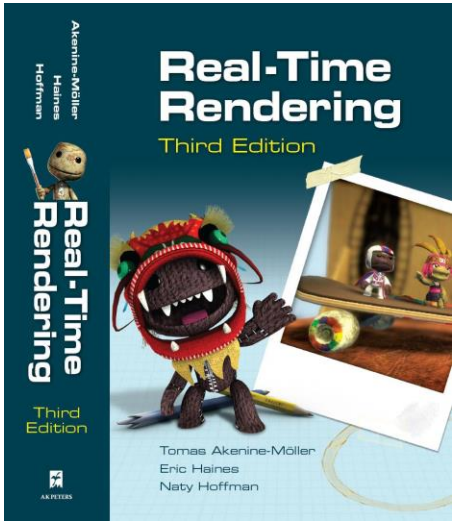
Resources

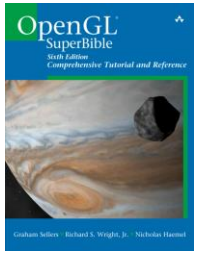
Graphics I (2D)

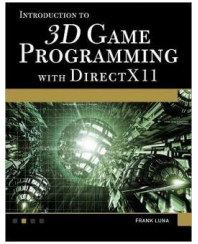




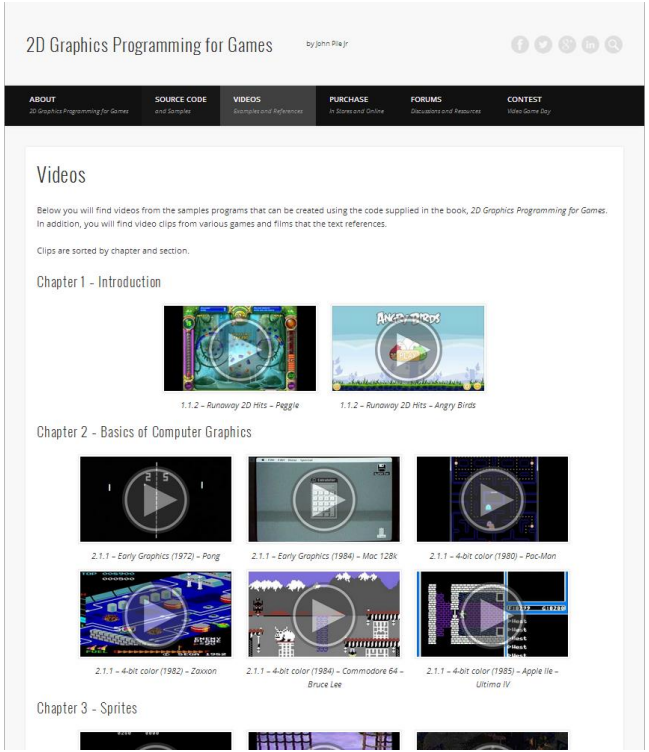
Graphics II (3D)

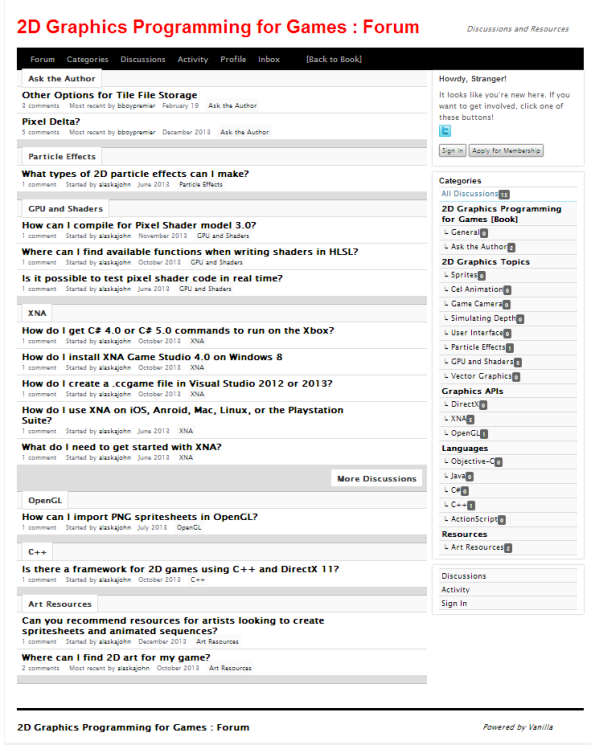






Further 2D Graphics Resources





Source Code

https://github.com/alaskajohn/2dGPfG

Or, download latest zip here.

Current Status as of July 2013:

C# / XNA	Complete
C++ / OpenGL	*Started*
C++ / DirectX 11	Not Started
Flash	Not Started

Additional APIs and Languages will be added regularly to the GitHub repo. If there is specific material you're looking for, please use the poll on our facebook page and I'll prioritize based on the results.

If you have specific questions, please post them in the 2D Graphics Programming for Games forum.

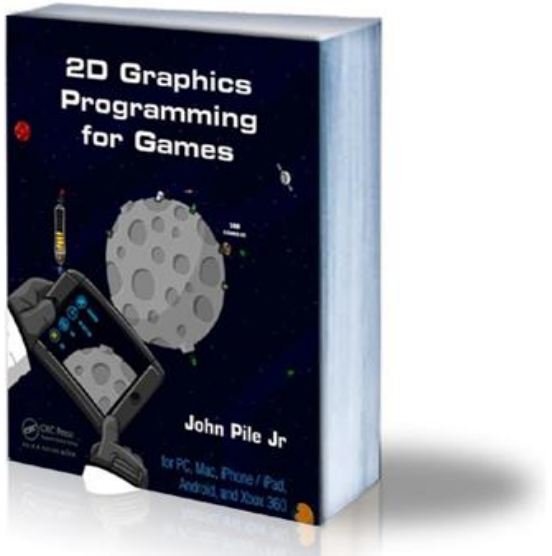
Share this:

* Syllabus

2D Graphics Programming for Games

AK Peters | CRC PRESS

Questions?



John Pile Jr

2dGraphicsProgramming.com

Twitter: @JohnPile

GDC EDUCATION SUMMIT

CHAMPLAIN COLLEGE 1878

GAME DEVELOPERS CONFERENCE 2014

SAN FRANCISCO, CA
MARCH 17–21, 2014
EXPO DATES: MARCH 19–21

Previous Curriculum

Year 1

Computer Theory
Game History & Development
Calculus

C++ Programming I
Game Technology
Matrices, Vectors, & 3D Math

Year 2

C++ Programming II
Intro to Networking & Security
Discrete Math

Game Architecture
Game Production I
Data Structures

Year 3

Graphics I (3D)
Linux / Unix
Computer Systems for Soft Eng

Graphics II (3D)
Game Production II
Elective

Year 4

Game Capstone
AI for Games
Networking for Games

Game Physics
Senior Production
OS Architecture

+ an additional 48 credits of core liberal arts

Updated Curriculum

Year 1

C++ Programming I
Game History & Development
Calculus

C++ Programming II
Game Technology
Matrices, Vectors, & 3D Math

Year 2

Graphics I (2D)
Data Structures
Discrete Math

Game Architecture
Game Production I
Intro to Networking & Security

Year 3

AI for Games
Linux / Unix
Computer Systems for Soft Eng

Graphics II (3D)
Game Production II
Elective

Year 4

Game Capstone
Networking for Games
Elective

Game Physics
Senior Production
OS Architecture

+ an additional 48 credits of core liberal arts