

ECE 595 / CS 491 / CS 591  
**Real-Time Rendering &  
Graphics Hardware**

Pradeep Sen  
Advanced Graphics Lab

Class 7  
February 7, 2007

### Announcements

- Intro project due Monday Feb 12 by midnight

### Last time

- Graphics engines
- Real-time graphics APIs (OpenGL)

### Today

- More on vertex and fragment programs
- Live coding!
- DirectX

### Vertex program

- Extension GL\_ARB\_vertex\_program

### Simple vertex program

```
!!ARBvp1.0
ATTRIB v              = vertex.position; # vertex position

# pass in the.mvp matrix
PARAM.mvp[4]         = { state.matrix.mvp };

# define the output
OUTPUT.oPos          = result.position # output vertex

# transform the vertex v by the.mvp matrix
DP4.oPos.x,.mvp[0],v;
DP4.oPos.y,.mvp[1],v;
DP4.oPos.z,.mvp[2],v;
DP4.oPos.w,.mvp[3],v;

END
```

### Simple vertex program

```
!!ARBvp1.0
ATTRIB v              = vertex.position; # vertex position

# pass in the.mvp matrix
PARAM.mvp[4]         = { state.matrix.mvp };

# define the output
OUTPUT.oPos          = result.position # output vertex

# temporaries
TEMP temp;

# transform the vertex v by the.mvp matrix
DP4 temp.x,.mvp[0],v;
DP4 temp.y,.mvp[1],v;
DP4 temp.z,.mvp[2],v;
DP4 temp.w,.mvp[3],v;

MUL oPos, temp, 9999;

END
```

### Per-vertex attributes (inputs)

- Some of the per-vertex inputs to the vertex program include:
  - vertex.position
  - vertex.weight
  - vertex.weight[n]
  - vertex.normal
  - vertex.color
  - vertex.fogcoord
  - vertex.texcoord, unit 0
  - vertex.texcoord[n], unit n
  - vertex.attrib[n]

### Vertex outputs

- The interpolant outputs of the vertex program (and the inputs to the fragment program) include:
  - result.position
  - result.color
  - result.fogcoord
  - result.texcoord, unit 0
  - result.texcoord[n], unit n

### Vertex program instruction set

ABS	v	v	absolute value
ADD	v,v	v	add
ARL	s	a	address register load
DP3	v,v	ssss	3-component dot product
DP4	v,v	ssss	4-component dot product
DPH	v,v	ssss	homogeneous dot product
DST	v,v	v	distance vector
EX2	s	ssss	exponential base 2
EXP	s	v	exponential base 2 (approximate)
FLR	v	v	floor
FRC	v	v	fraction
LG2	s	ssss	logarithm base 2
LIT	v	v	compute light coefficients
LOG	s	v	logarithm base 2 (approximate)
MAD	v,v,v	v	multiply and add

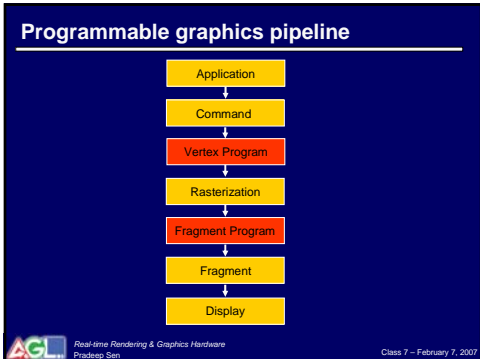
AGL Real-time Rendering & Graphics Hardware Pradeep Sen Class 7 - February 7, 2007

### Vertex program instruction set

MAX	v,v	v	maximum
MIN	v,v	v	minimum
MOV	v	v	move
MUL	v,v	v	multiply
POW	s,s	ssss	exponentiate
RCP	s	ssss	reciprocal
RSQ	s	ssss	reciprocal square root
SGE	v,v	v	set on greater than or equal
SLT	v,v	v	set on less than
SUB	v,v	v	subtract
SWZ	v	v	extended swizzle
XPD	v,v	v	cross product

AGL Real-time Rendering & Graphics Hardware Pradeep Sen Class 7 - February 7, 2007

- ### Vertex program binding and loading
- Live demo
- AGL Real-time Rendering & Graphics Hardware Pradeep Sen Class 7 - February 7, 2007



- ### Fragment program
- Often known as "pixel shaders"
  - Extension GL\_ARB\_fragment\_program
- AGL Real-time Rendering & Graphics Hardware Pradeep Sen Class 7 - February 7, 2007

### Simple fragment program

```

!!ARBfp1.0
MOV result.color, [1, 0, 0, 0];
END
  
```

AGL Real-time Rendering & Graphics Hardware Pradeep Sen Class 7 - February 7, 2007

- ### Per-fragment inputs
- Also known as interpolants
  - Same list as before (output of vertex program)
- AGL Real-time Rendering & Graphics Hardware Pradeep Sen Class 7 - February 7, 2007

- ### Fragment program outputs
- Only two outputs:
    - result.color
    - result.depth
- AGL Real-time Rendering & Graphics Hardware Pradeep Sen Class 7 - February 7, 2007

### Fragment program instruction set

ABS	v	v	absolute value
ADD	v,v	v	add
CMP	v,v,v	v	compare
COS	s	ssss	cosine with reduction to [-PI,PI]
DP3	v,v	ssss	3-component dot product
DP4	v,v	ssss	4-component dot product
DPH	v,v	ssss	homogeneous dot product
DST	v,v	v	distance vector
EX2	s	ssss	exponential base 2
FLR	v	v	floor
FRC	v	v	fraction
KIL	v	v	kill fragment
LG2	s	ssss	logarithm base 2
LIT	v	v	compute light coefficients
LRP	v,v,v	v	linear interpolation
MAD	v,v,v	v	multiply and add

AGL Real-time Rendering & Graphics Hardware Pradeep Sen Class 7 - February 7, 2007

## Fragment program instruction set

MAX	v, v	v	maximum
MIN	v, v	v	minimum
MOV	v	v	move
MUL	v, v	v	multiply
POW	s, s	ssss	exponentiate
RCP	s	ssss	reciprocal
RSQ	s	ssss	reciprocal square root
SCS	s	ss--	sine/cosine without reduction
SGE	v, v	v	set on greater than or equal
SIN	s	ssss	sine with reduction to $[-\pi, \pi]$
SLT	v, v	v	set on less than
SUB	v, v	v	subtract
SWZ	v	v	extended swizzle
TEX	v, u, t	v	texture sample
TXB	v, u, t	v	texture sample with bias
TXP	v, u, t	v	texture sample with projection
XPD	v, v	v	cross product



Real-time Rendering & Graphics Hardware  
Pradeep Sen

Class 7 – February 7, 2007

## Live demo!

- Programming of various shaders



Real-time Rendering & Graphics Hardware  
Pradeep Sen

Class 7 – February 7, 2007