# EE 4702-1, GPU Programming

## When / Where

- Here (149 EE Building), MWF 13:30-14:20 Fall 2015

- http://www.ece.lsu.edu/koppel/gpup/

## Offered By David M. Koppelman

- Room 345 ERAD

- Office Hours: Monday-Friday 15:00-16:00

## Prerequisites by topic:

- C++.

# GPU Definition

*GPU*: Graphics Processing Unit

- Runs 3D graphics in place of CPU...

  ... because it's much better at it.

- Also runs scientific-style computation in place of CPU.

GPU is Main Component of Video Cards

Major Companies and Brands

- NVIDIA

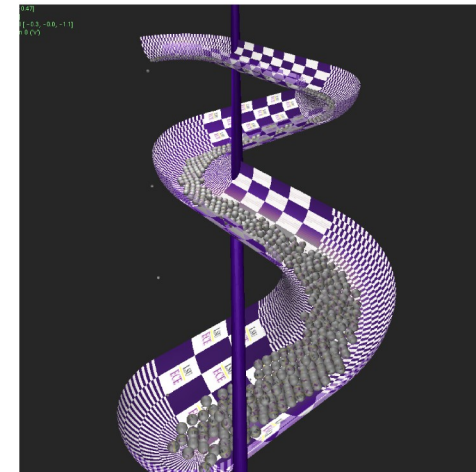- ATI (Compaq/HP)
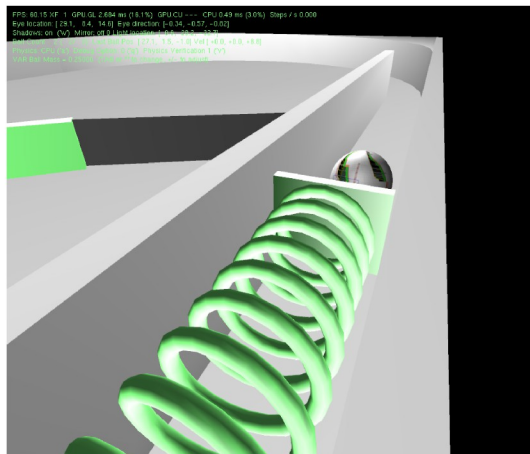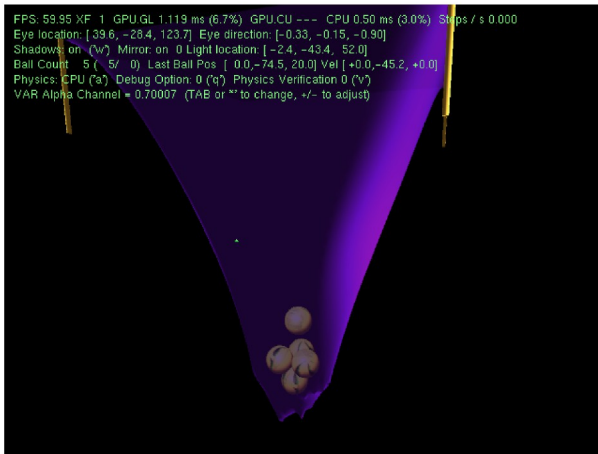
# This Course

**Focus is on GPU programming**

- *Shader* programming with *OpenGL Shader Language* (*OGSL).*

- *GPGPU* programming with *CUDA.*

**Also Some 3D Graphics, Game Physics**

- Will cover enough graphics, OpenGL, and CUDA to do cool stuff.

**Game Physics Term Project**
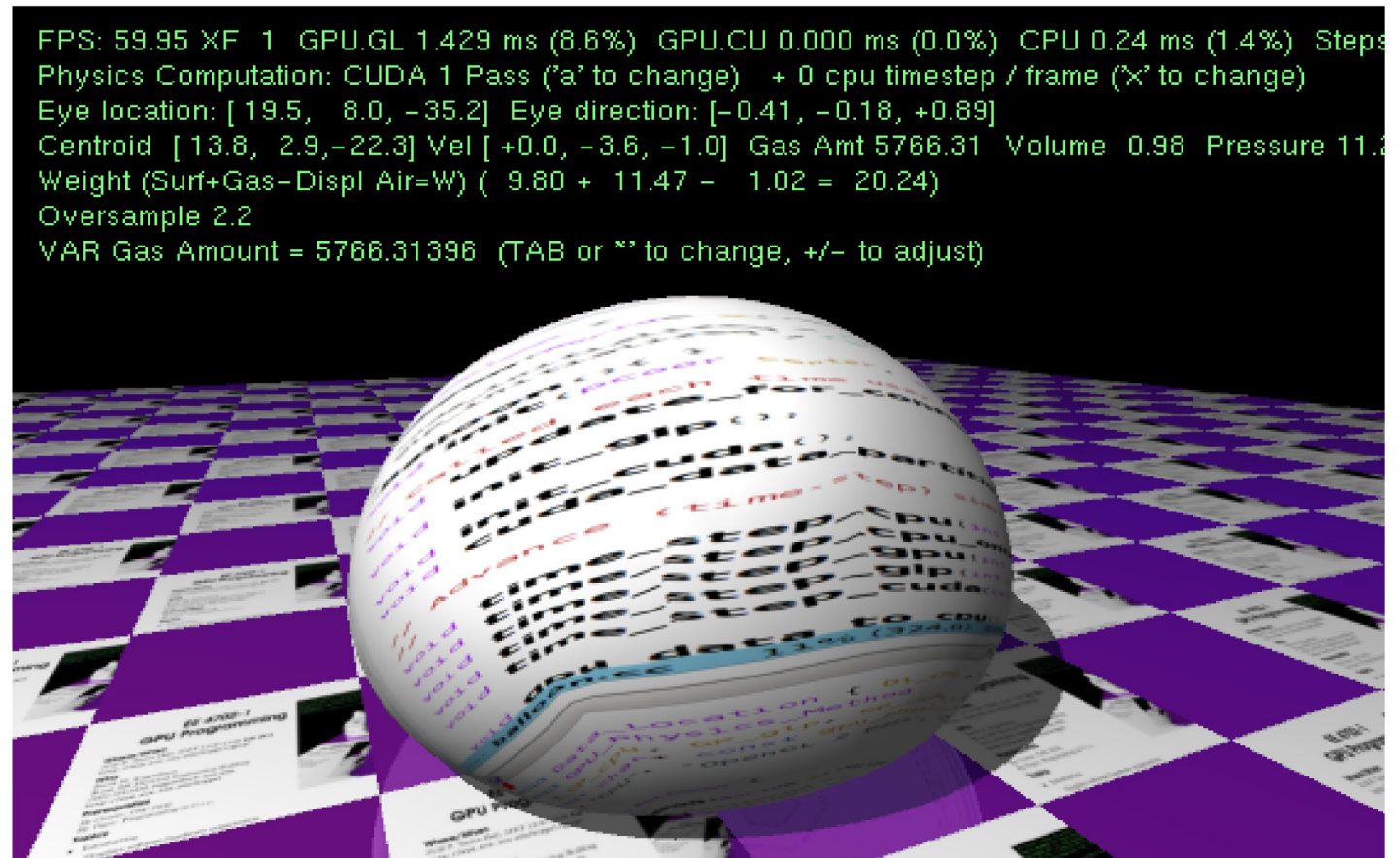
**Past Student Project Screenshots:**

# Balloon Demo

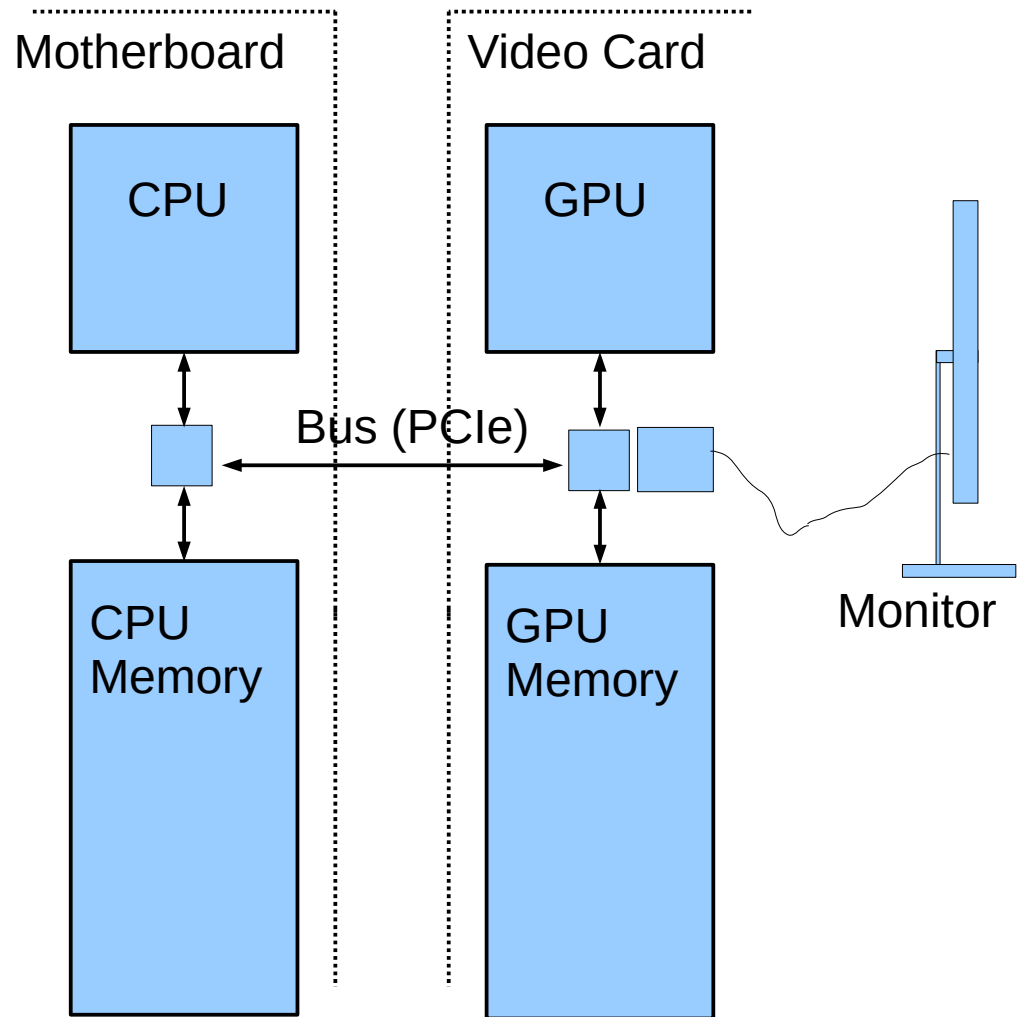Simulation of a balloon.

GPU always runs 3D graphics.

Code can switch between CPU-only and CPU/GPU physics.

Term project can be an extension of this code.

# System Overview

Quick look at how the GPU fits into the larger system...

Motherboard

Video Card

CPU

GPU

Bus (PCIe)

CPU
Memory

GPU
Memory

Monitor

# System Overview: Hardware

**Typical Hardware**

- On Computer motherboard: CPU, CPU Memory

- On Video Card GPU, GPU Memory

- Connection between CPU/GPU via Bus, e.g., PCI Express (PCIe).

- Connection from video card to monitor.

Motherboard

Video Card

CPU

GPU

Bus (PCIe)

CPU Memory

GPU Memory

Monitor
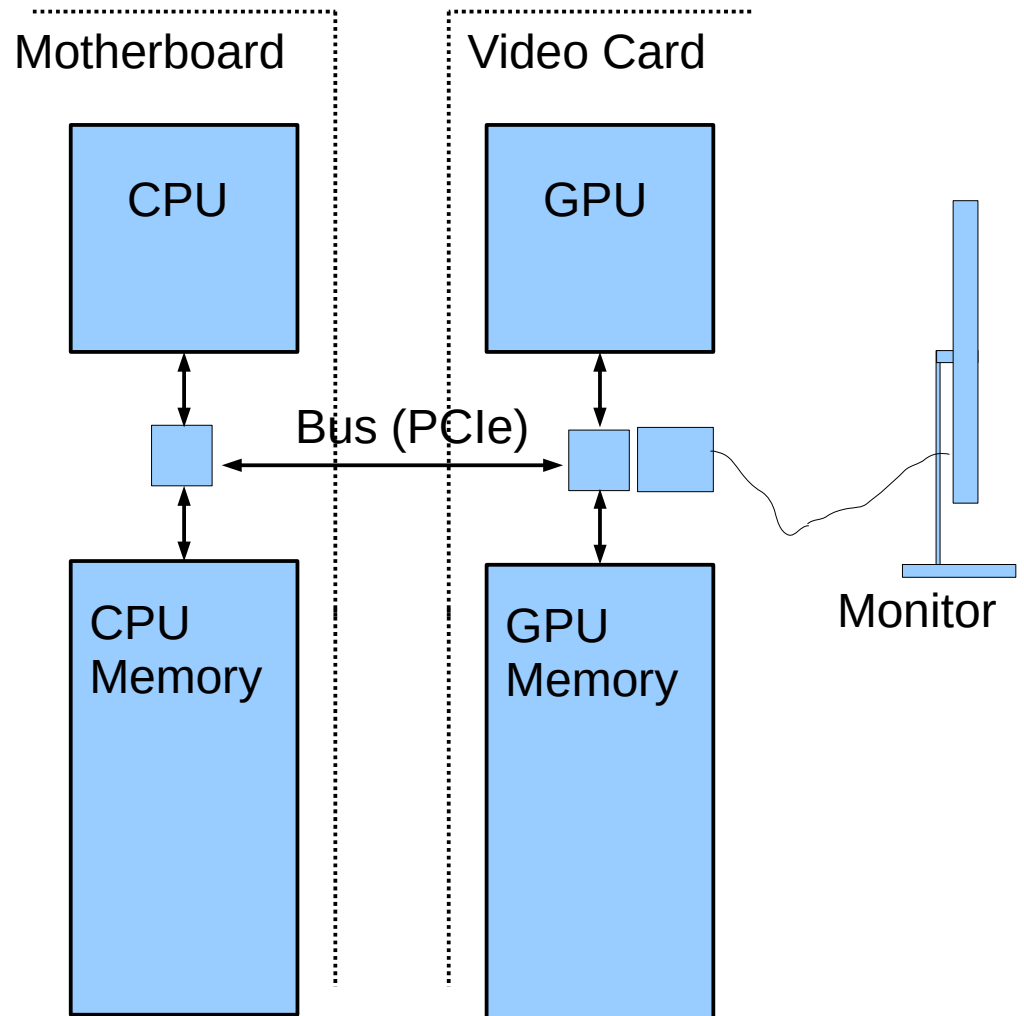
# System Overview: Frame Buffer

*Frame Buffer*

- Area of memory continuously converted to a video signal.

- Simple mapping from memory address to pixel coordinate.

Older Systems

- Frame buffer in CPU memory.

- Application programs wrote frame buffer directly.

Typical Current Systems

- Frame buffer in GPU memory.

- Frame buffer written by GPU hardware (typical)...

    … at end of long chain of events initiated by application.

# System Overview: Frame Buffer

## Frame Buffer Contents

- Position in FB corresponds to particular pixel on display.

- In illustration, first FB element is upper-left pixel.

- A common FB element size is 32 bits.

- Frame buffer format varies with video mode and of course system.

Motherboard

Video Card

CPU

GPU

Bus (PCIe)

CPU Memory

GPU Memory

Frame Buffer

Monitor
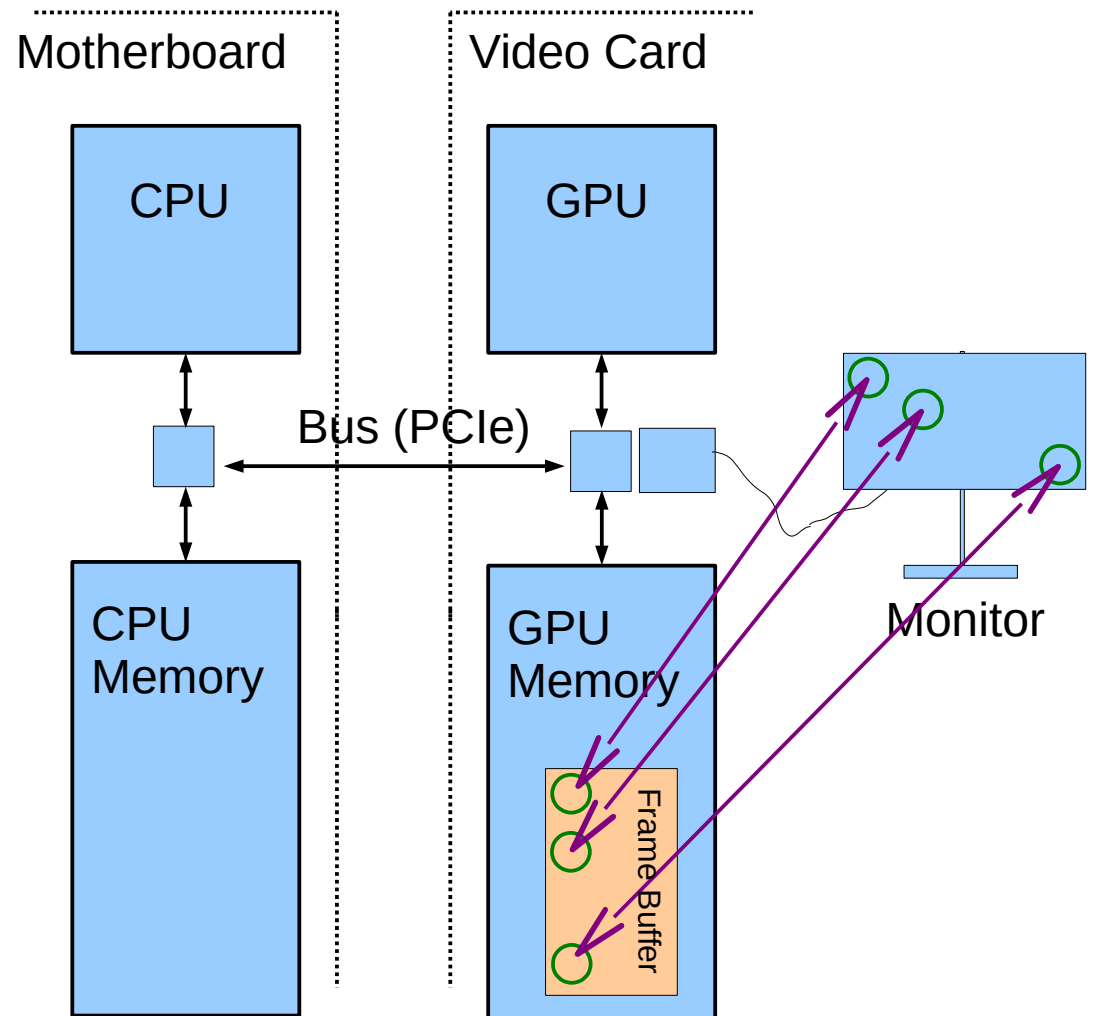
# Simple Frame Buffer Code Example

## Consider Code

- frame_buffer[10][20] = 0;

## For Simple Code Example Assume

- The frame buffer is in CPU memory.

- Array frame_buffer points to the frame buffer location.

- Writing a "1" to the array makes a pixel white.

- Writing a "0" to the array makes a pixel black.

# Simple Frame Buffer Example

// Make screen all white. (Assume 1 is white.)

//
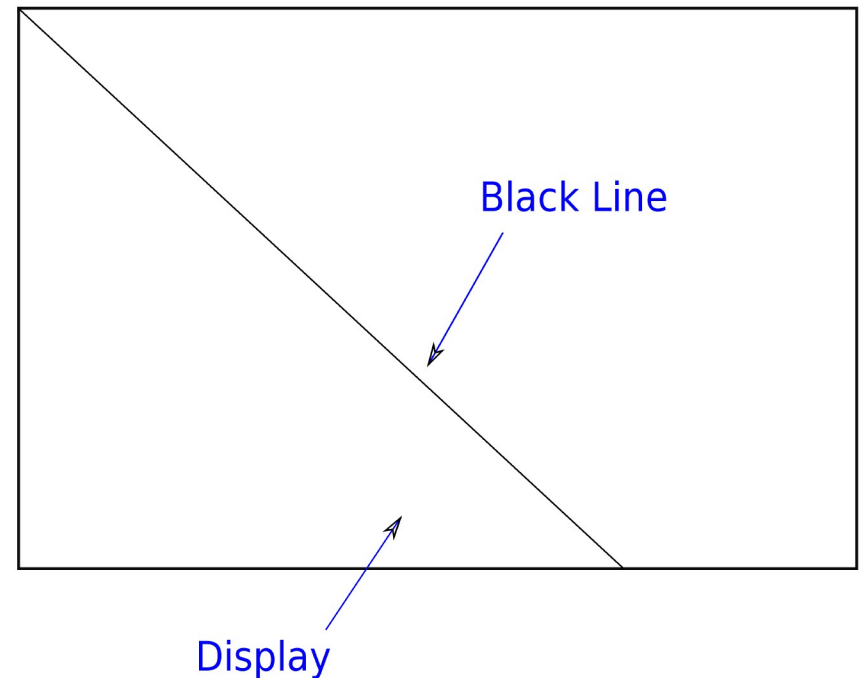
for ( int x = 0;  x < width;  x++ )

  for ( int y = 0; y < height; y++ )

    frame_buffer[x][y] = 1;


// Draw a black diagonal line.

//

for ( int x = 0; x < height; x++ )

  frame_buffer[x][x] = 0;

Black Line

Display

# GPU Rationale

Code on prior slide looked simple.

How much more complex would balloon code be?

Alot, of course!

That's half of the motivation for a GPU:

- Graphics (3D animated, especially) requires alot of computation.
- CPU is less suited for that kind of computation.

# System Overview: Code Organization

Simplified View of Code on Typical System

Who Wrote Code?

- User. (Call it the Ap.)

- Vendor of GPU. (Call it the driver.)

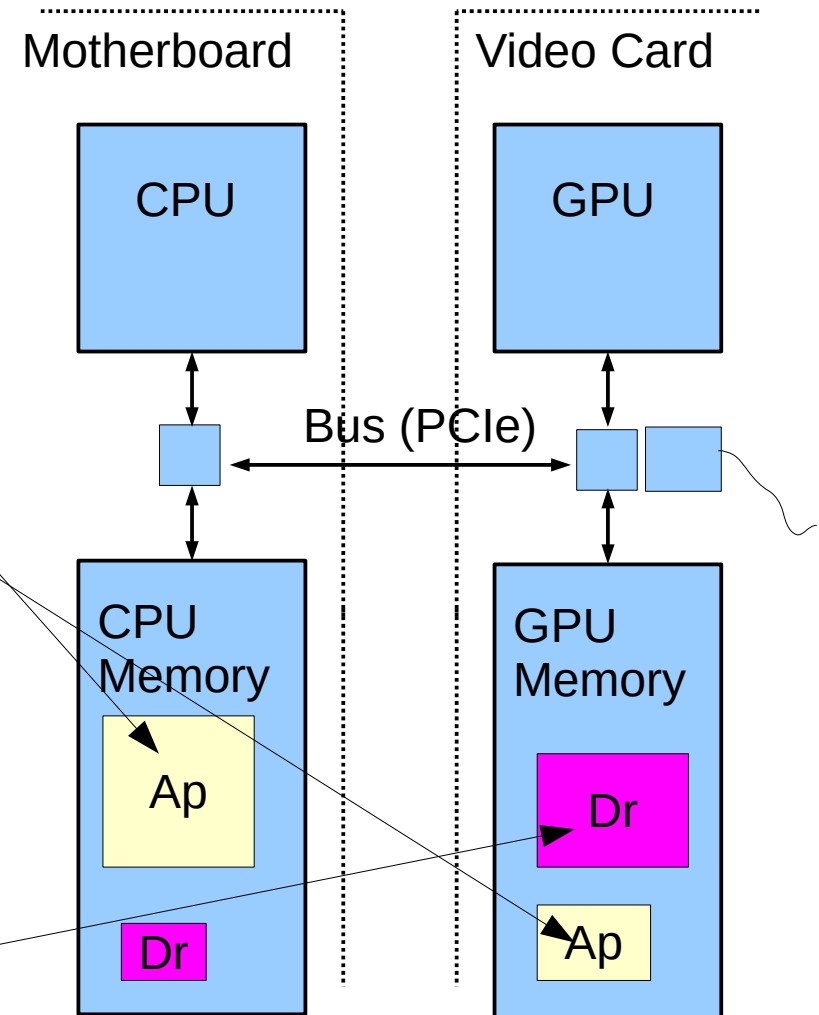Where Does It Run?

- CPU

- GPU

All Four Possible:

- User/CPU, User/GPU, Vendor/CPU, Vendor/GPU

# System Overview: Software

Just For Today, Oversimplify to
Two Kinds of Software

- *Application Program* (Ap for short*)

  – Written by ap. programmer.

  – E.g.,  Balloon Demo

  – Most of Ap runs on CPU.

  – Part of Ap may also run on GPU.

- *GPU Driver* (Driver or Dr for short)

  – Written by GPU manufacturer.

  – E.g., NVIDIA 185.18.14

  – Driver code runs on both CPU and GPU.

  – Most work done by driver code that runs on GPU.

**Motherboard**

**Video Card**

CPU

GPU

Bus (PCIe)

CPU Memory

Ap

Dr

GPU Memory

Dr

Ap

*Don't confuse Ap with App.

# System Overview: Running of Application

Typical Execution

- Application, running on CPU, ready to emit next frame.

- App. calls driver on CPU...

    ...driver on CPU starts more driver code on GPU...

    ...application resumes on CPU (while GPU driver code still running).

- Process above repeated many times for a frame.

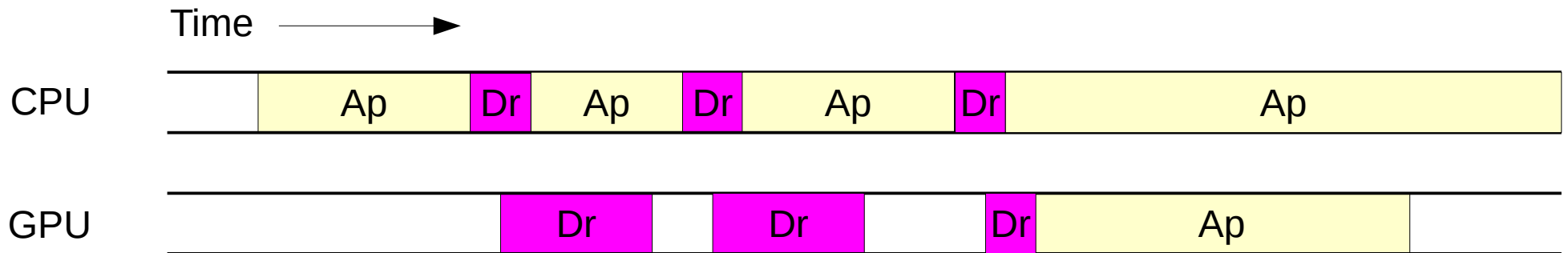- Driver code ultimately will write frame buffer.

Important Points

- CPU and GPU can run code at same time.

# System Overview: Execution Activities

Example Below

- Ap calls driver three times.

- First two times, driver does all work, mostly on GPU.

- Third time that ap calls driver, driver starts some ap code on GPU.

Time →

CPU   | Ap | Dr | Ap | Dr | Ap | Dr | Ap |

GPU   | Dr | Dr | Dr | Ap |

# Course Coverage

Emphasis: GPU coding for high performance.

Topics Needed For Term Project (a dynamic simulation)

Topics

- Rudiments of Animation by Dynamic Simulation

- Term Project

- 3D Graphics basics: coordinates, transforms, primitives, colors, textures.

- Data movement and staging, efficiency.

- Coding with GPU *shader* model, CPU/GPU load balancing.

- Coding with CUDA, GPU physics.

# Graphics Equipment

For assignments, use equipment in Workstation Lab

Workstation Lab

- Room 126 EE Building

- Several kinds of computers.

- This course uses the coolest ones.

Graphics Workstations

- Alienware and More-Serious Dell Machines

- Best three: dual Kepler GPUs.

- Other dual GPU machines.