Lecture 16: Introduction to XNA

Prof. Aaron Lanterman
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dungeon Quest

• Developed in 4 days at the 2007 GDC at the XNA contest
• By Benjamin Nitschke and Christoph Rienaecker

Installing XNA Game Studio Express

• Go to creators.xna.com, click on “Education” then “Getting Started”
• XNA only works with the Express Edition of Microsoft Visual C#
  – Various versions of Visual Studio seem to coexist OK
• Follow instructions under “Setting Up XNA Game Studio Express”
  – Don’t skip the “Visual Studio 2005 Service 2005 Express Editions Service Pack 1” step
  – Be sure to get the “Refresh” version of XNA Game Studio Express
• Many things described here may change when Version 2.0 comes out

Careful if you’re on Windows x64

• XNA normally targets “AnyCPU”
• Will break when you try to run on x64 machines, since x64 versions XNA framework .dlls don’t exist (yet)
• Workaround: Change PlatformTarget of app to x86
  – Use a text editor to add this line to each PropertyGroup item of your csproj file in a text editor:
    – <PlatformTarget>x86</PlatformTarget>

Info from Benjamin Nitschke’s blog, “Getting XNA to work in Windows XP x64,” Sunday, August 27, 2006, exdream.no-ip.info/blog/PermaLink.aspx?guid=89629a7-3185-438d-852a-ed0f197716c2
Paradigm shift

- XNA is built on top of DirectX 9
  - Not built on MDX or Managed DirectX
- DirectX 9 has a fixed function pipeline, but XNA doesn’t!
  - Everything done with shaders


Why no fixed-function pipeline?

In Microsoft’s own words (paraphrased):

- Programmable pipeline is the future
  - Neither Direct3D 10 or Xbox 360 have fixed-function pipeline
- Early adopters and customers said cross-platform goal more important than fixed-function pipeline
- Fear is someone would start and finish their game using the fixed-function APIs, and then get dozens of errors when they tried to compile it on the Xbox 360
- Better to know your code works on both right from the beginning


Hello Bluescreen

```java
public class SampleGame : Game {
    private GraphicsComponent graphics;

    public SampleGame() {
        this.graphics = new GraphicsComponent();
        this.GameComponents.Add(graphics);
    }

    protected override void Update() {
    }

    protected override void Draw() { this.graphics.GraphicsDevice.Clear(Color.Blue);
        this.graphics.GraphicsDevice.Present();
    }

    static void Main(string[] args) {
        using (SampleGame game = new SampleGame()) {
            game.Run();
        }
    }
}
```


Some convenient things about XNA

- Don’t need to mess with Win32-ish boilerplate (opening a window, etc.)
- Easy interfacing with the Xbox 360 controller (for both Windows and Xbox 360)
- Storage (“saved games”) unified between Windows and Xbox 360
  - On Xbox 360, have to associate data with a user profile, put on hard drive or memory card, etc.
  - XNA “emulates” this on windows

XNA strengths & weaknesses: audio

- Uses XACT, Microsoft’s Audio Creation Tool
- Nice for modifying sound effects directly in XACT until you like them
- No support for .mp3 or .wma - all you can plug in are uncompressed wave files
  - 10M game can turn into 100M
- Xbox 360 has XMA, and Windows has ADPCM, but game still 3-5 bigger than it might otherwise be

Info from Alistair Wallis, "Microsoft XNA: A Primer," interview with Benjamin Nitschke
www.gamecareerguide.com/features/328/microsofts_xna_a_.php?page=4

Caveats about Xbox 360 development

- Many TVs cutoff 5-10% of the pixels around the edge
  - Keep text & important info away from there
- Xbox 360 handles post processing and render targets a little differently than the PC

Info from Alistair Wallis, "Microsoft XNA: A Primer," interview with Benjamin Nitschke
www.gamecareerguide.com/features/328/microsofts_xna_a_.php?page=4