Addressing the Need for a Java Game Curriculum

Scott A. Wallace
Washington State University Vancouver
swallace@vancouver.wsu.edu

Andrew Nierman
University of Puget Sound
anierman@ups.edu

Computer Games
- In 2004, game sales reached $7.4 billion
  - More than doubled since 1996
- NES
  - Released in 1985 (USA)
  - Around for many students’ memorable lifetime

Games in Academia
- Various Types of Institutions
  - Colby College
  - University of Michigan
  - Universiteit Utrecht (The Netherlands)
- Various Levels of Curriculum
  - Capstone to CS1 through Master’s level

Why Computer Games?
- Pander to student interests
- Everyone else is doing it
- Leverage student’s interest
  - Increased project relevance
  - Increased retention (perhaps)
- Perfect platform for synthesis projects
  - Networking, algorithms, AI, software engineering
Our History

- A special topics course in Computer Game Design offered at WSU Vancouver in Fall '05
  - Not well advertised
  - 4 students (seniors, masters students)
  - Various levels of motivation (some – some more)
  - Perfect dry run

- Game engine borrowed from Colby College
  - Not in active development
- No textbook
  - Seemed to make students quite nervous

Lessons & Questions

- Game engine is important
  - Provides foundation for less motivated students
  - Highly motivated students can go beyond the engine
  - Can provide the basis for class discussions
    - Should be consistent with text

- Were our students really interested?
  - How could we support them more appropriately
    - Especially marginally motivated students...

Our Schools

- WSUV
  - Junior / Senior / Masters
  - Non-traditional students
  - Becoming four year...
- UPS
  - Small liberal arts setting
  - Where to add games?

Assessment of Interest

<table>
<thead>
<tr>
<th></th>
<th>Projects in 1st/2nd year</th>
<th>2nd/3rd year course</th>
<th>3rd/4th year course</th>
<th>vs. other CS courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>6.7</td>
<td>8.8</td>
<td>8.8</td>
<td>8.7</td>
</tr>
<tr>
<td>Stdev</td>
<td>2.9</td>
<td>1.7</td>
<td>1.6</td>
<td>1.8</td>
</tr>
<tr>
<td>N</td>
<td>31</td>
<td>32</td>
<td>32</td>
<td>30</td>
</tr>
</tbody>
</table>

0: no interest       10: extreme interest
The interest is there…

- Next, survey engines to support course goals
- Want to create better Computer Scientists
  - Not “entertainment industry specialists”
  - Synthesis vs. new material
- Want to support student learning outside the classroom

Language Considerations

- C++ is most industry relevant language
- UPS and WSUV curriculum centers on Java
- High school AP exam is in Java
- Java much more prevalent in CS1/CS2
- For highly motivated students this is secondary, but not so for less motivated ones

2D / 3D Considerations

- 3D is the industry relevant choice
- Both UPS and WSUV offer junior/senior graphics courses on openGL & 3D graphics
  - Don’t want a duplicate class
- What is the emphasis for a games class?
  - 2D graphics are more accessible
  - Upper level students will have more time for other features

Our Desires

- Language: Java
- 2D/3D Focus: 2D
- Intended Audience: Students
- Cost & Licensing: Low, or None
- Source Code Availability: Yes, Open
- Documentation: Yes
Engines in Academia

- Torque Engines
  - C++/Torque script, 2D/3D, $50/seat with source
- DXFramework (U. Michigan)
  - C++, 2D (3D), open source, minimal docs
- SAGE (U. North Texas)
  - C++ 3D, beta docs & tutorials, unlicensed source
- GEDI (Marist College)
  - C++, 2D, early in development, unlicensed source

Where are the Java Engines?

- Turning to Google…

<table>
<thead>
<tr>
<th>DevMaster database</th>
<th>Java games middle-ware</th>
<th>top 100 &quot;java game engine&quot;</th>
<th>Active Java engines</th>
<th>Active 2D Java engines</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 (10)</td>
<td>4 (12)</td>
<td>4 (7)</td>
<td>14</td>
<td>2</td>
</tr>
</tbody>
</table>

The Candidates…

- Genuts
  - Aimed at mobile devices
  - Limited infrastructure for more sophisticated apps
  - Limited documentation and tutorials
  - Free, but no source code
- GTGE
  - Suitable for more advanced games
  - Documentation and tutorials evolving (still new)
  - Free, but no source code

An Unfilled Niche

- We are collaboratively building JIGE
  - Java implementation
  - 2D emphasis
  - Open source
  - Academic focus, multi-campus development

  Intended to support a dedicated class
  Intended to support isolated game projects
Supporting Existing Courses

- Modular architecture
- Want to be able to ‘pop in’ new features
  - To the engine itself
  - To demonstration games

- Course projects
  - We create initial game
  - Students create isolated plug in module/class
  - Don’t need to go into ‘game programming’
  - But the project is visually representative of task and engaging

Current Status

- Not quite ready for prime time
  - Basics implemented, but demos are not rich

- Applied for NSF CCLI grant in mid May
  - Work redirected from engine to grant

- WSU Pullman students led grass roots effort to get game class via teleconferencing system
  - This year’s course is taught in a combination of languages
  - Java, C++ (DXFramework)
  - Intend to incorporate this experience into JIGE

Jige Physics Demo

JIGE -- Graphics

- Abstraction of graphics & drawing primitives
  - Image classes
  - Offscreen buffers
  - Coordinate system transforms

- Multiple back ends via abstract factory
  - Java 2D
  - Open GL (via LWJGL)
JIGE -- Physics

- Two supplied physics engines
  - Simple (sprite level physics)
  - More complex ('real' physics model)
    - Based on Erin Catto's 2006 GDC presentation
    - Provide physical laws of game environment
    - Detect collisions between objects
- Designed to be redesigned (by students)
  - This is a primary design consideration for the engine as a whole

Supporting Synthesis

- Dedicated classes offered at WSUV
- Planned for UPS
- Engine provides foundation for student work
- Expected that some students will modify it
- Audience: late sophomores to seniors