Physics based animation

Lecture 01 - Introduction

Grégory Leplâtre

g.leplatre@napier.ac.uk, room D32
School of Computing
Edinburgh Napier University

Semester 1 - 2016/2017

Objectives

- Programme of the module

Objectives

- Programme of the module
- Administration

Outline

1 Introduction

2 Assessment

3 References

Physics-based animation - context

- Animation techniques

Physics-based animation - context

- Animation techniques
  - By hand (keyframe animation)

Physics-based animation - context

- Animation techniques
  - By hand (keyframe animation)
  - Data driven (motion capture)

Physics-based animation - context

- Animation techniques
  - By hand (keyframe animation)
  - Data driven (motion capture)
  - Procedural
    - non-physically based
    - physically-based

Relevant topics

- Rigid/soft bodies
- Particles
- Fluids
- Hair
- Cloth

Module aims

- Gain experience in physics-based simulation development, particularly the technical aspects (e.g., achieving real-time performance)

Module aims

- Gain experience in physics-based simulation development, particularly the technical aspects (e.g., achieving real-time performance)
- Examine some of the fundamental techniques used in physics animation development such as the underpinning maths, performance bottlenecks, particles, collisions, constraints, numerical issues

Module aims

- Gain experience in physics-based simulation development, particularly the technical aspects (e.g., achieving real-time performance)
- Examine some of the fundamental techniques used in physics animation development such as the underpinning maths, performance bottlenecks, particles, collisions, constraints, numerical issues
- Put together various physics-based simulations

Learning Outcomes

Learning Outcomes	Supported by	Assessed by
LO1: Develop insight into the design and implementation process that occurs within modern physics-based animation software development.	Lectures & Practicals	Coursework
LO2: Consider the different physics-based techniques.	Lectures, practicals & tutorials	Coursework & class tests
LO3: Produce a physics-based simulation utilising a development environment by applying relevant design and development processes.	Practicals	coursework
LO4: Demonstrate a practical understanding of the fundamental mathematics underpinning physics based development	Tutorials	Class tests

Timetable

	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00
Mon						SET09119.L2 Dr Gregory Leplatre Physics Based Animation Lecture Merch.G9 Weeks: 2-13			
Tue								SET09119.P1b Mr Sam Serrels Physics Based Animation Practical Merch.B56.co Weeks: 2-13	
Wed	SET09119.T1a Dr Gregory Leplatre Physics-Based Animation Tutorial Merch.H11 Weeks: 2-13	SET09119.T1b Dr Gregory Leplatre Physics-Based Animation Tutorial Merch.H11 Weeks: 2-13	SET09119.L1 Dr Gregory Leplatre Physics Based Animation Lecture Merch.H10 Weeks: 2-13						
Thu									
Fri	SET09119.P1a Mr Sam Serrels Physics Based Animation Practical Merch.B56.co Weeks: 2-13								

Structure

- Lectures:
  - physics principles, design, technology, and algorithms
- Practicals:
  - Development of physics-based simulations
  - Clear, computationally fast, elegant code (beyond prints)
  - Use of Visual Studio as a development environment
- Tutorials:
  - Fundamental mathematical principles
  - Three class tests

Work plan

- You WILL have to work around **12-13 hours a week** on the module:
  - 5 hours contact (lectures, practical, tutorial)
  - 7-8 hours self study
- Coursework will require some organisation outside class time
  - Use the Games Lab, it's what it is there for
- Keep up with practical work! There is one every week
  - We won't be pausing - so falling behind at any stage means it will be difficult to catch up

Working through practicals

- Run by Sam Serrels (s.serrels@napier.ac.uk)
- Regularly check in your work to **BitBucket**
- Make videos of your working practical simulations
  - put the course title, simulate on title and your name at the start of the video
- Write modular, readable code
- Experiment

Equipment and resources

- Games lab equipped with high-end PCs
  - 8am-9pm weekdays, 9am-5pm week-ends
  - All software you need is there
- JKCC has Visual Studio installed
- Visual Studio available from Dreamspark for home use
- Using your own wired physical controller is an option to be considered

Outline

1 Introduction

2 Assessment

3 References

Coursework

- **Develop a physics-simulation**
  - Must be buildable on the PCs in the Games Lab
  - Can use any control mechanism you like
- Expand on what you learn during the module (e.g., cloth animation control)
- Three parts
  1. Pitch presentation (week 5)
  2. Design document (week 8)
  3. Implementation, report and demonstration (week 14)

Report submissions

- Reports have a purpose (technical document)
- Use SIGGRAPH template
- Submit PDF (not docx, txt, bmp, png)
  - Name & Student Number (First Page)
  - Title
  - Page Numbers
  - Sections (Introduction, Overview, Summary)
  - Figures (captions and referenced in text)
  - Tables (captions and referenced in text)
- Naming convention: LastName_FirstName_Matric_ReportName.pdf
- See past projects/reports `http://games.soc.napier.ac.uk/physics.html`

Class tests

- Mathematics covered in tutorials is assessed in three class tests
  - Week 6: Vectors, matrices, transformations, FK, IK, etc
  - Week 10: Motion, derivatives, Newton's laws, projectile motion, etc
  - Week 13: Energy, momentum and collisions, rotational motion, etc

Outline

1 Introduction

2 Assessment

3 References

Key points

- **Game Physics Engine Development**, by David Bourg (2001)
- **Game Physics**, by David Eberly (2010)
- **Real-Time Collision Detection**, by Christer Ericsson (2004)
  - Essential for the collision detection part of the module

To do this week

- Take a look at last year's materials to form a better idea of the contents of the module.
- Take a look at past projects (link from Moodle)
- **Bring pen and paper to every class**.
- **Read Tutorial 1 notes for Wednesday**