Module webpage:

- Lecture Slides
- Labs
- Readings from the course

Submission and feedback
- TBC
LEARNING OUTCOMES

On Completion of this course, Students should be able to:

- Outline and describe the architecture and components of a physics-based motion pipeline that would be employed in a typical game.
- Discuss and explain basic rigid body dynamics and the mechanics of deformable systems.
- Analyze and compare different approaches for collision detection and collision response of rigid and deformable objects.
- Build a real-time rigid body simulation system for convex polyhedra.
- Discuss state-of-the art issues in real-time physics in the IET industry.
RELEVANT REFERENCES

Baraff, Witkin and Kass “Physically based Animation Course Notes” – Siggraph 2001
  • http://www.pixar.com/companyinfo/research/pbm2001/

Mueller, James, Theurey, Stam “Real-time Physics” – Siggraph 2008
  • http://www.matthiasmueller.info/realtimephysics/index.html

Kenny Erleben - “Physics based animation”
David H.Eberley - “Game Physics Engine Development”

Ian Millington – “Game Physics Engine Development”
Eric Lengyel – Mathematics for 3D Game Programming and Computer Graphics
ASSESSMENT

100% based on coursework
- ~55% Labs
  - Weekly demos
  - Final Cumulative submission
    - Including: Video, written report, Source code
- ~40% Final Assignment
- ~5% End of Term Assessment

IMPORTANT NOTE: All work is expected to be done individually. Where other help or sources are used these should be duly referenced. Students are expected to familiarise themselves with and abide by college’s regulations on plagiarism.

Unless otherwise stated a penalty of 20% per day will be incurred for late submission
01: INTRODUCTION TO PHYSICALLY BASED MODELLING

18/01/2016
A form of Procedural Modelling

Synthesize motions of inanimate objects due to external forces

Motivations:
- Realism / plausibility
- Automation
- Interaction
APPLICATION AREAS

- Virtual reality; design prototyping
- Assembly planning
- Interactive computer animation
- Surgical simulation; preoperative planning
- Computational robotics; manipulation
- Video games
- Scientific visualization
- Education
LucasArts’ Star Wars - The Force Unleashed [2008] featured 3 different state-of-the-art simulation technologies in industry:

- Havok Physics: for gameplay rigid body dynamics physics
- NaturalMotion Euphoria for physically responsive non-player character behaviour
- Pixelux DMM for deformable and destructable effects physics

Tech Trailer: http://www.gametrailers.com/video/unleashing-the-star-wars/29964
THE FORCE
SOME PHYSICS ENGINES
STATE OF THE ART (RESEARCH)

There is a vast amount of research literature in Physically Based Animation that deals with highly complex effects that is not yet fully exploited in games.
TYPES OF ANIMATION

- **Procedural**
  - Rule-based
  - Automated
  - Synthesised

- **Scripted**
  - Manually authored
  - Interpolated key frames

- **Data Driven**
  - Performance Capture

Predator animation is © Adriano Rinaldi [http://web.tiscali.it/maya_tutorial/]

CS7057: REALTIME PHYSICS (2015-16) - JOHN DINGLIANA PRESENTED BY MICHAEL MANZKE
Animation is encoded in a set of rules. Mathematical models are employed to define motions.

Procedural Animation

- Behavioural Animation
- Physically-based Animation
- Ad-hoc Procedural Animation

- Particle Systems
- Rigid Bodies
- Deformable Objects
- Fluids

Discussed in Real-time Animation or Autonomous Agents

i.e. Anything not explicitly physically based e.g. splines
PHYSICALLY BASED ANIMATION

**Kinematics**: describes motion without considering forces that cause it

- **Inverse Kinematics**: given positions, find orientations
- **Forward Kinematics**: given orientations, find positions

- **Inverse Dynamics**: given the start and end points (i.e. the motion), what are the forces
- **Forward Dynamics**: given the start conditions and forces, calculate the motion

**Dynamics**: describe motion taking into account mass and forces

Most relevant for interactive apps
Rules of physics used to calculate motion

Take into account...
- Forces
  - e.g. Friction, Gravity, Buoyancy, Elasticity
- Collisions / spatial occupancy
MOTIVATIONS

**Why use Physically based models?**

**Inherits advantages from procedural animation**
- Non-linear experiences: unlimited possibilities from relatively small amount of manual content creation
- Non-deterministic

**Extends procedural animation with real-world physical constraints**
- Relatively well defined (even mathematically)
  - Discrete representations exist - can encode this in programs
  - Finite set of rules and parameters
- Plausible (hopefully)
Relatively small number of basic concepts are re-used for most game physics:

- Newton’s laws of motion
- Stress and Strain: deformable models
- Navier Stokes equation: fluids

Major challenge is in finding numerical solutions that are sufficiently complex and sufficiently generalizable
**CLASSES OF GAME PHYSICS**

- **Off-line Physics**
  - e.g. Cinematics / cutscenes
  - Pre-calculated
  - Pre-rendered

- **Pre-computed Physics**
  - Canned Physics
  - Secondary Animations e.g. some character animations

- **Faked Physics**
  - Geometric procedures resembling Physics
  - e.g. some water animations

- **Effects Physics**
  - Live Secondary Animations
  - One-way Interaction
  - e.g. Particle effects, cloth

- **Gameplay Physics**
  - Fully interactive
  - Affects the user
EFFECTS PHYSICS

Havok FX Demo © 2005 Havok
GAMEPLAY PHYSICS

1958

1962

1972

Scorched Earth

THE INCREDIBLE MACHINE

TRESPASSER

λ²

CUT the ROPE

ANGRY BIRDS
IN-GAME PHYSICS

**Efficiency:**
- on-the-fly real-time computation

**Plausibility:**
- not necessarily accurate or real, but convincing

**Fidelity:**
- model reasonably complex phenomena to make it worthwhile

**Robustness:**
- run unsupervised over many frames without exploding
BASIC ANIMATION LOOP
BASIC PHYSICS LOOP

Rendering

Collision Handling

Simulation
BASIC PHYSICS LOOP (2)

- Apply forces and update state of object based on forces
- Check if objects are intersecting
- Apply collision forces/impulses
# PHYSICS STATE VARIABLES & CONSTANTS

<table>
<thead>
<tr>
<th>Position</th>
<th>Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity</td>
<td>Acceleration / Forces</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Orientation</th>
<th>Moments of Inertia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial Occupancy</td>
<td>Centre of Mass</td>
</tr>
<tr>
<td>Angular Velocity</td>
<td>Torque</td>
</tr>
<tr>
<td>Plasticity</td>
<td>Elasticity</td>
</tr>
<tr>
<td>Density</td>
<td>Viscosity</td>
</tr>
<tr>
<td>Opacity</td>
<td>Texture</td>
</tr>
<tr>
<td>Friction</td>
<td>...</td>
</tr>
</tbody>
</table>
Collision Detection and Intersection testing

check if an object intersects another – a common operation in computer graphics
A diverse problem (lots of special cases) but not extremely difficult to solve e.g. Lots of Intersection Source Code Available Here:
http://www.geometrictools.com/LibFoundation/Intersection/Intersection.html
PROBLEM: COMPLEXITY

Current graphical data sets are generally more detailed than we wish to deal with for collision detection.

Possible solution: Use simplified representation. (Even this is generally overkill for game physics)
COLLISION PROXIES

e.g. Hit boxes commonly used in games approximate the actual mesh.
COLLISION PROXIES

Havok Example:
- All physics objects are approximations of display meshes.
- Finer Rag-doll collision proxy for character used in effects physics interactions with objects in scene.
- A much coarser bounding ellipsoid proxy used for navigation (yellow blob below)

Images © Havok 2007
DYNAMICS PROXIES
TWO PHASE COLLISION DETECTION

Acceleration strategy: cull trivial cases (discussed in more detail later in the module)

- **Broad phase**
  - Per-object: coarse culling of non-colliding cases

- **Narrow phase**
  - Pair-wise between objects
  - Per-feature
  - Narrow-in on collision – calculate finer features e.g. collision manifold, penetration depth
BOUNDING VOLUMES

Use a coarse over approximation of the object i.e. A bounding volume (conservative overestimate) that is geometrically much simpler than the object. Only do more detailed intersection tests if bounding volume intersects.

Types of bounding volumes: sphere, axis-aligned bounding box (AABB), oriented bounding box (OBB), eight-direction discrete oriented polytope (8-DOP) and convex hull.

Image from Real-time Collision Detection – by Christer Ericson
Bounding Volume Hierarchies

Hierarchical, multi-resolution representations of objects.

- Enables Progressive Culling:
  - Starting at root, only check children if this collides
  - Repeat until we get to finest level then do triangle checks
Once objects are found to be colliding collision process applies forces/impulses to stop them intersecting to create plausible physically based reactions.

Key Concerns in Interactive Physics:

- Robustness: response must assure system remains stable in all possible states
- Plausibility: response to collision must be believable
- Efficiency: physics processing must be fast enough to enable real-time frame rates
FLUIDS

Images © Ron Fedkiw (http://physbam.stanford.edu/~fedkiw/)
## TENTATIVE LECTURE PLAN

<table>
<thead>
<tr>
<th>MON</th>
<th>THURS</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 Jan</td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td>Particle Systems</td>
</tr>
<tr>
<td>25 Jan</td>
<td>LAB: RIGID BODIES</td>
</tr>
<tr>
<td></td>
<td>Collision Detection</td>
</tr>
<tr>
<td>1 Feb</td>
<td>LAB: BROAD PHASE</td>
</tr>
<tr>
<td></td>
<td>Collision Detection 2</td>
</tr>
<tr>
<td>8 Feb</td>
<td>LAB: NARROW PHASE</td>
</tr>
<tr>
<td></td>
<td>Collision Response</td>
</tr>
<tr>
<td>15 Feb</td>
<td>LAB: COLLISION RESPONSE</td>
</tr>
<tr>
<td></td>
<td>Bounding Volume Hierarchies</td>
</tr>
<tr>
<td>22 Feb</td>
<td>LAB: marking</td>
</tr>
<tr>
<td></td>
<td>Beyond Rigid Bodies</td>
</tr>
</tbody>
</table>

Reading Week 29 February – 4 March

| 7 Mar     | FINAL PROJECT MEETINGS     |
|           | Multi-collisions           |
| 14 Mar    | Deformable Objects, Cloth, Deformable Collisions |
| 21 Mar    | Fluid Simulation, Eulerian Models |
| 28 Mar    | Easter Monday              |
|           | Destruction/Fracture, Characters Physics, Hair |
| 4 Apr     | FINAL ASSIGNMENT PRESENTATIONS |
PLANNED ASSESSMENT

Weekly Labs
- Particle Systems ~10%
- Rigid-body Animation: ~10%
- Broad-phase Collision Detection: ~10%
- Narrow-phase Collision Detection: ~10%
- Collision Response: ~10%

Cumulative Lab Submission ~5%
- Rigid body engine & particle system
  - Youtube Video
  - Report
  - Source code (for the record: not marked based on this)

Final Project: ~40%
End of Term Assessment: ~5%