

Gentle Introduction to Physics in Games and Demos

Jetro Lauha
Seminar presentation

Assembly 2005, 28th - 31st July 2005

Contents

- Basic components of physical simulation
- Integration to the application
- Physics featured in demoscene releases and case study of *Stair Dismount* and sequels
- Questions & Answers

Basic Components of Physical Simulation

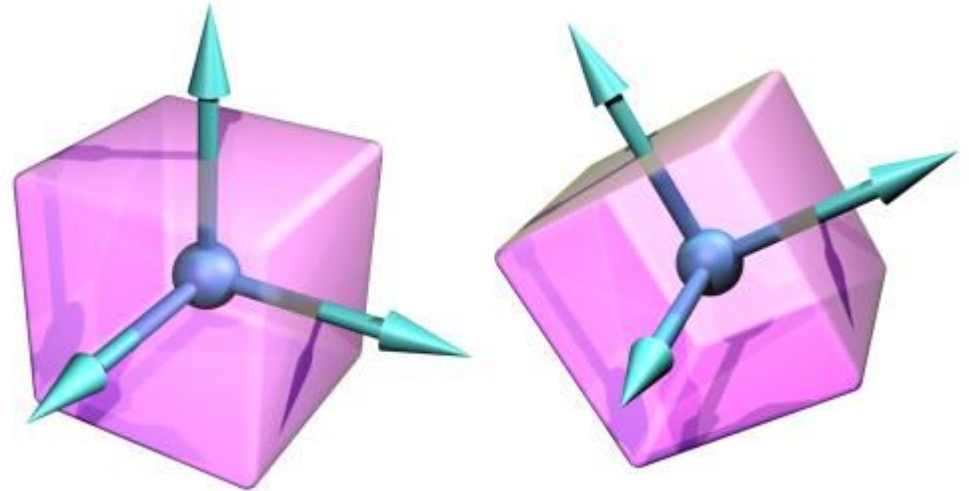
- Kinematics
- Rigid body dynamics
- Constraints and joints
- Collision detection

Kinematics

- Single particle
 - Mass
 - Position
 - Velocity
 - Applied forces (e.g. gravity)
- Can be joined together with springs or constraints
 - Jakobsen's Verlet-system as an example (later)

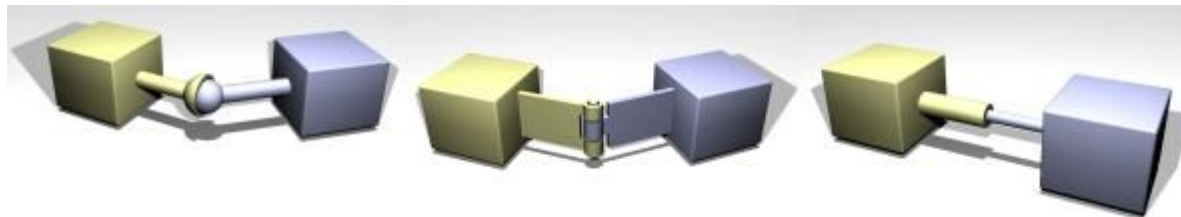
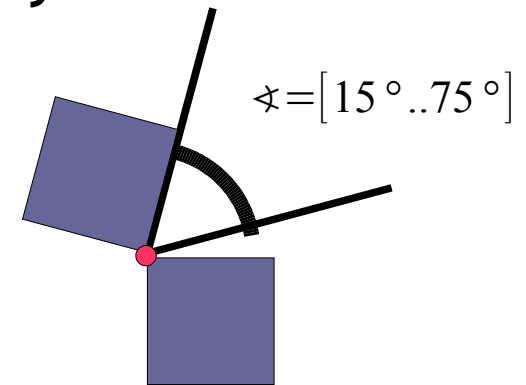
Rigid Body Dynamics

- Extension to particle physics
 - Orientation
 - Angular velocity
 - Shape
 - Center of mass
 - Inertia tensor
- Forces applied to arbitrary point
 - Relative to center (typically center of mass)



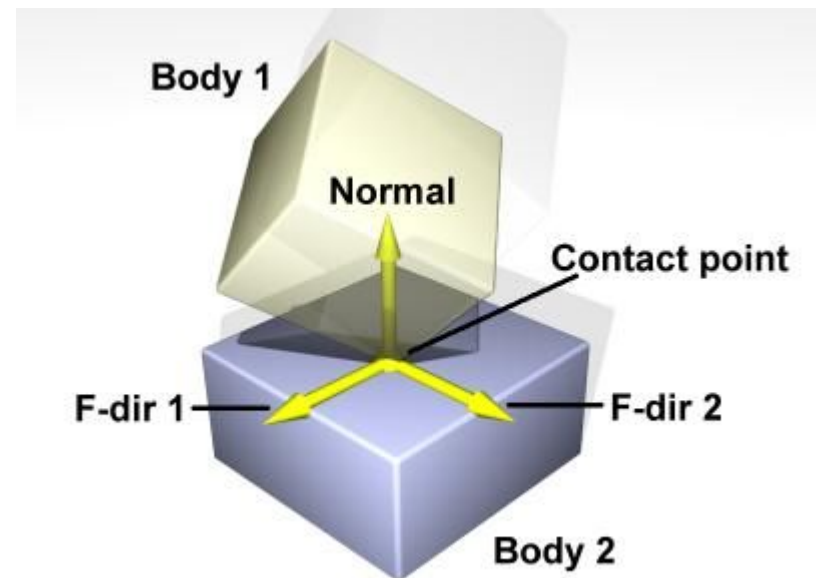
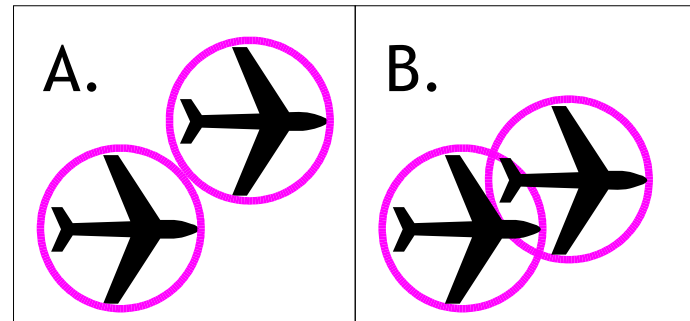
Constraints and Joints

- Joints limit degrees of freedom
 - position: 3, orientation: 3 - collectively 6
- Constraints, examples:
 - limit position to given region
 - force position to given plane
 - limit angle of hinge joint between given minimum and maximum angle



Collision Detection

- Coarse test
- Detailed contact
 - position
 - normal vector
 - penetration depth
- Collision handling
 - body & surface material properties (bounciness, slippiness) => coefficient of restitution, friction forces



Integration to the Application

- Proxy geometries
- Updating simulation
- Use of existing packages (Middleware)
- Verlet integration [Jak01]
 - Short look at one specific physics implementation technique

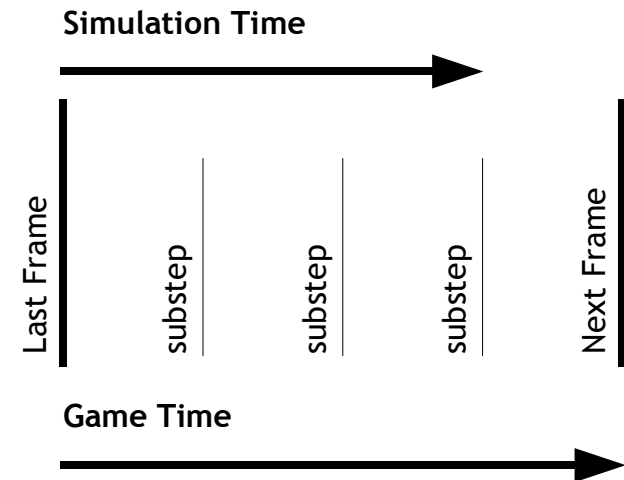
Proxy Geometries

- Primitive geometries
 - approximate given part of original detailed model
- Physics simulator's understanding of the model
 - Rendering code handles modification of detailed model to match with the proxy geometries (orientations, skinning)



Updating Simulation

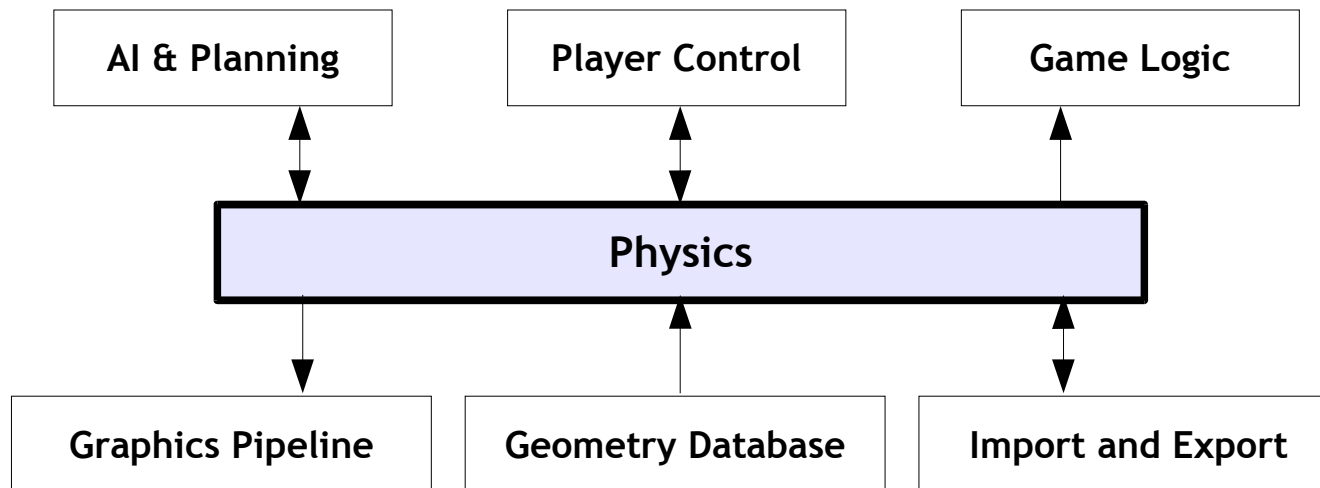
- Game applications contain several logical clocks
 - updating of game logic
 - physics simulation
 - rendering
- Use fixed size time step for physics simulation
 - Despite of several papers recommending dynamic one



[McL03, adaptation]

Use of Existing Packages (Middleware)

- Saves implementation time
- Saves development costs
- Applicability to be carefully evaluated



Verlet Integration

[Jak01]

- Simple, fast, relatively stable
- Single particles, clothes, plants, ragdolls
- Rigid bodies
 - Combined from particles with constraints
 - Simple basic building blocks used to create more complex systems
- Used in *Hitman*
 - Developer by IO Interactive, published by Eidos

Verlet Integration

[Jak01]

- Euler integration

$$x = x + v \cdot \Delta t$$

$$v = v + a \cdot \Delta t$$

- Verlet integration

$$x' = 2x - x^* + a \cdot \Delta t^2$$

$$x^* = x$$

$$F = ma$$

x^* previous position

$$2x - x^* = x + (x - x^*)$$

$$x - x^* \sim v$$

Verlet Integration

[Jak01]

```
#define DAMPING      (0.999)
#define TIMESTEPSQ  (0.02 * 0.02)

...

// pos,oldPos = position and previous position
// accumForces = combined forces affecting
//              the particle

Vector3 pos, oldPos, accumForces;

...

pos += DAMPING * (pos - oldPos) +
       accumForces * TIMESTEPSQ;
```

Verlet Integration

[Jak01]

Particles too close each other



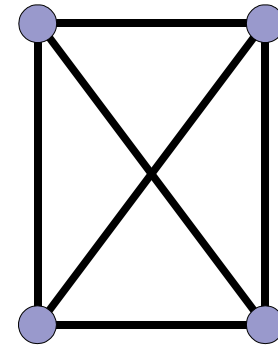
Constrained to fixed distance



Particles too far from each other



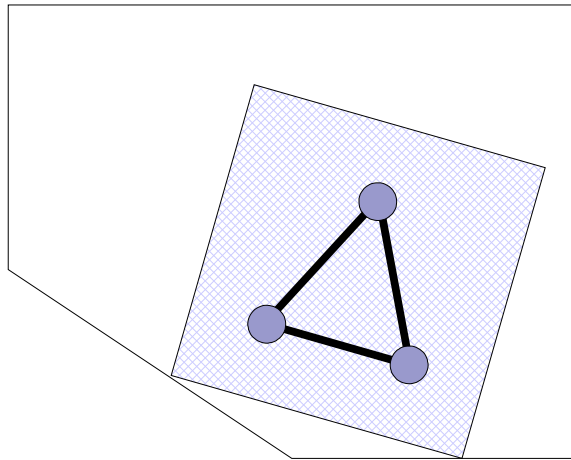
Body built of particles and constraints



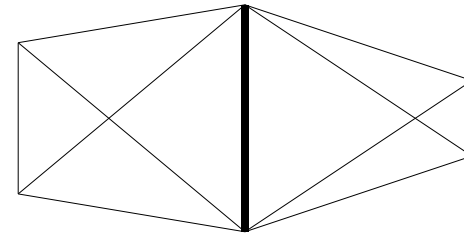
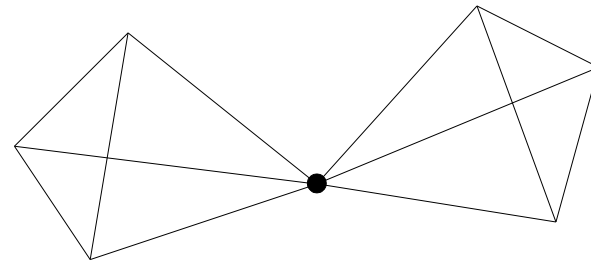
Verlet Integration

[Jak01]

Separate collision system
with particle system used
to define body rotation



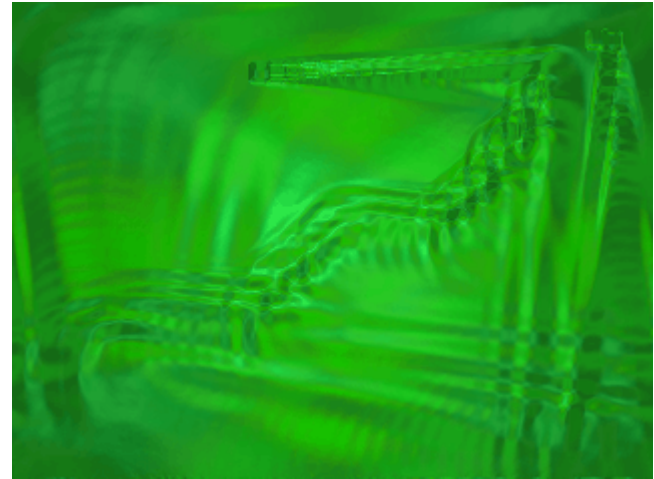
Pin joint (ball-and-socket)



Hinge joint

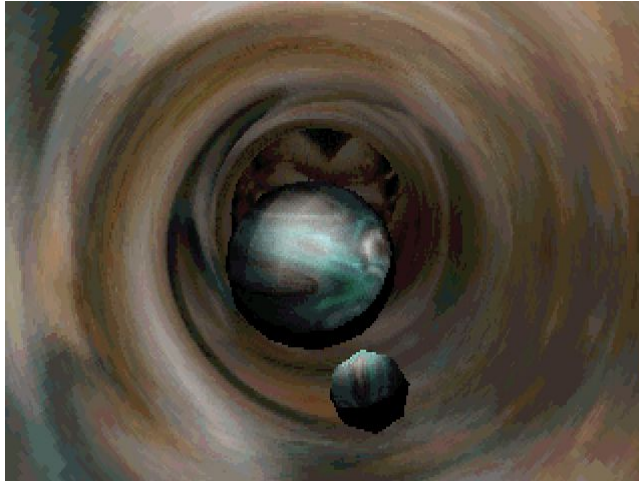
Physics Featured in Demoscene Releases

- Some demos featuring physics simulation
- Case study: *Stair Dismount* and sequels

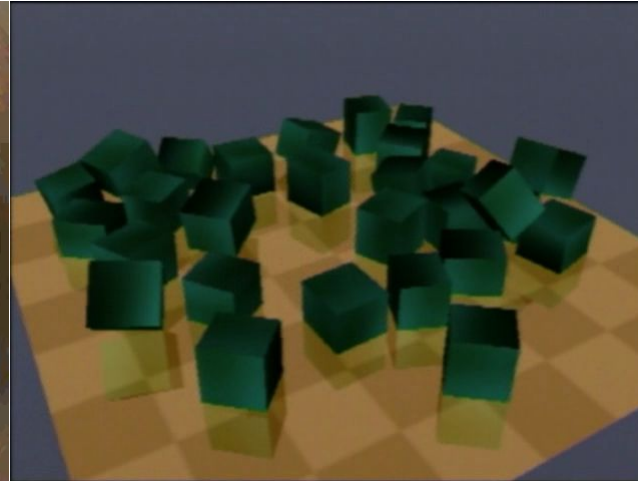


Some Demos Featuring Physics Simulation

Trauma:
Mindtrap
08/1997

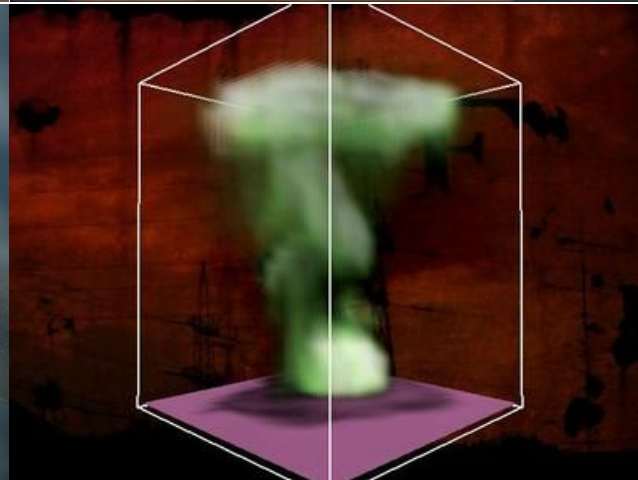
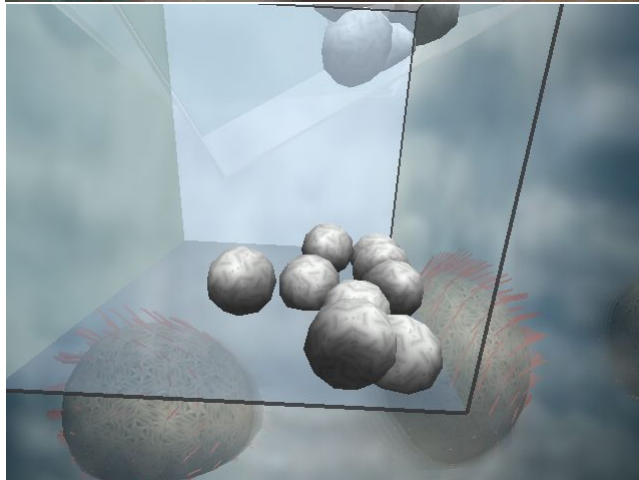


tAAt:
*Laatukauraa -
Quality Oat*
08/2002



PlayStation 2

Floppy:
Dream Equation
10/2002
&
Dream Equation II
07/2003



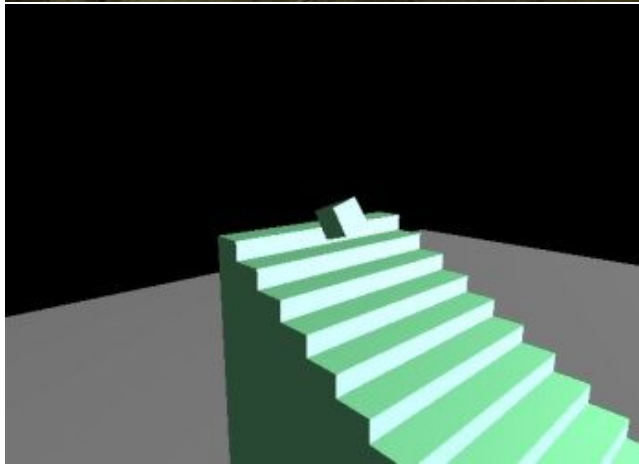
Faktory:
Feed your machine
08/2003
&
47'111.0
08/2004

Some Demos Featuring Physics Simulation

Fairlight:
Digital Dynamix
08/2003



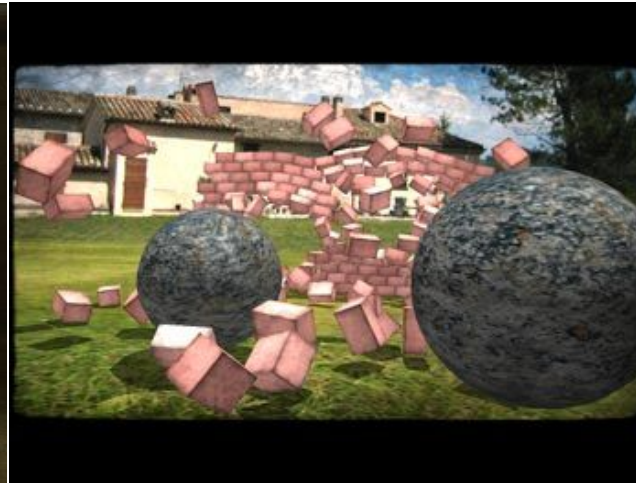
Lonely Coders:
Cubic Revolution
08/2004



4 KB intro

Screenshot from pouet.net

Screenshot from pouet.net



Mayoneez and
the boys:
MOPED
08/2004



Nesnausk!:
in.out.side:
the shell
05/2005

Screenshot from pouet.net

Case Study:

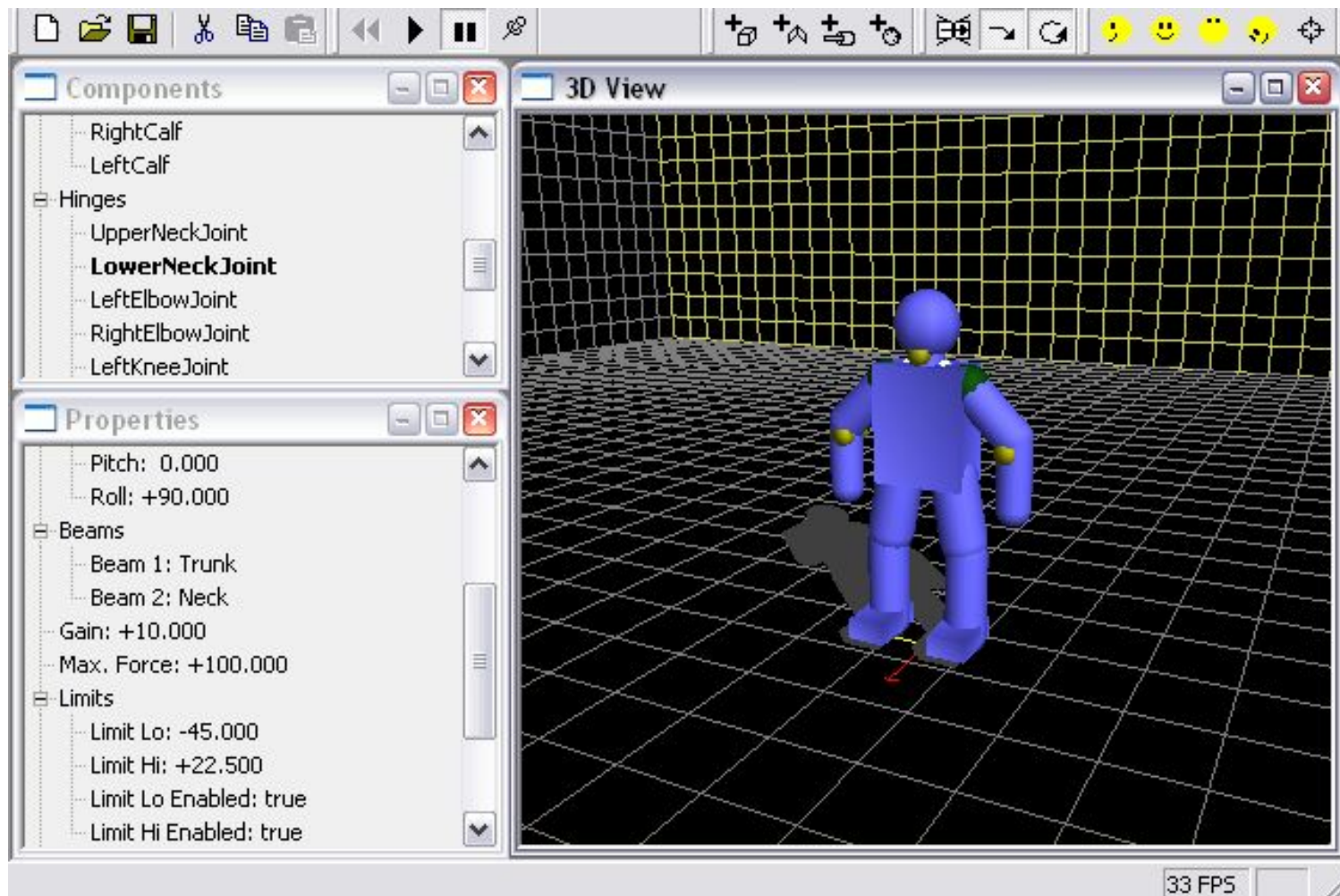
Stair Dismount and Sequels

- **Porrasturvat (Stair Dismount) - Assembly'02**
 - Stairs and a ragdoll
- **Rekkaturvat (Truck Dismount) - Assembly'03**
 - Truck and the ragdoll, mini editor (ramps etc.)
- **Dismount Levels (Preview) - Assembly'04**
 - Generic editor, integrated scripting language
 - Still in development (looking for contributors)
 - Has small community

Case Study: *Stair Dismount* and Sequels

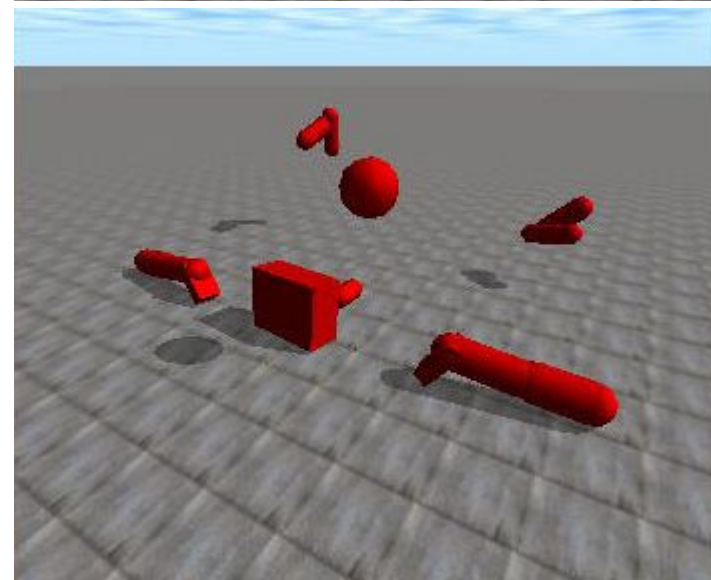
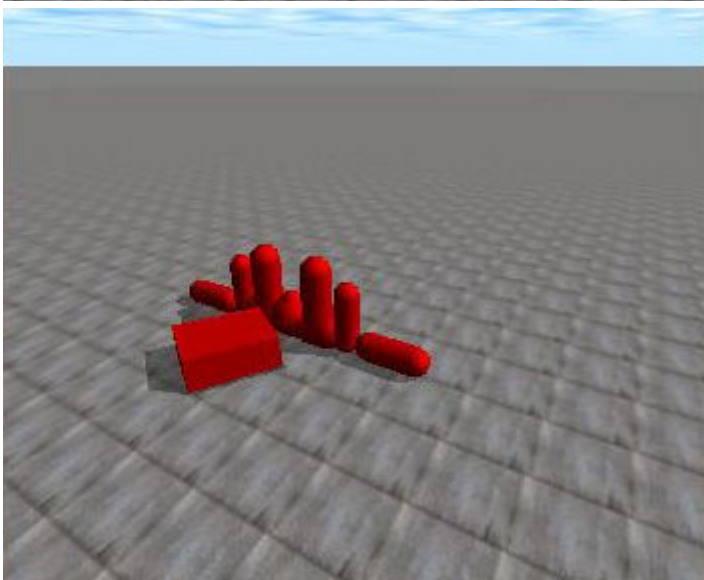
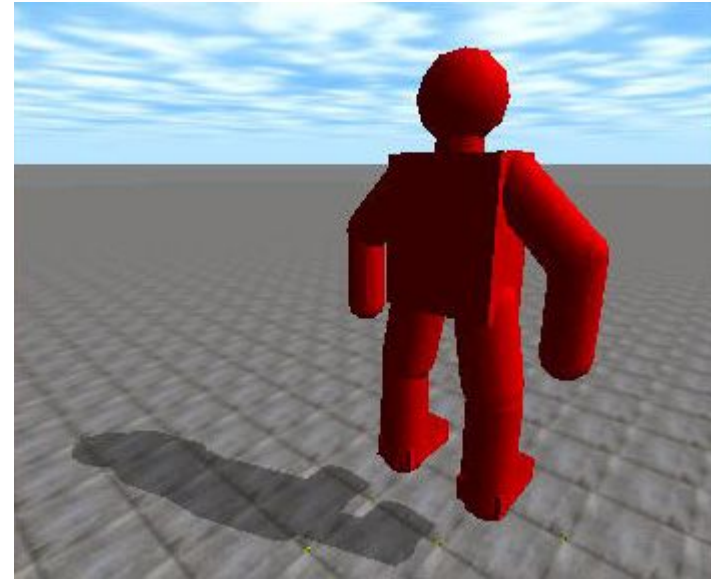
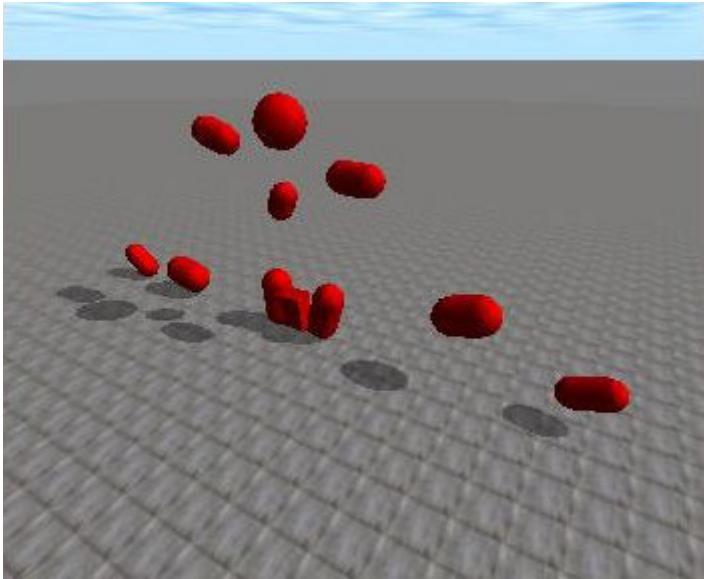
- *Juice* used to model ragdoll
- Use of open source libraries have saved a lot of effort and time with development
 - SDL, SDL_image, SDL_mixer, FMOD, zlib, CFL, libpng, libjpeg, ODE, libcurl, expat, libogg, AngelScript, TinyXml, Mersenne Twister
 - Turska - limited but simple framework/UI library as spin off from the games, features-added-as-needed
<http://turska.sourceforge.net> (v0.1.1)

Case Study: *Stair Dismount* and Sequels

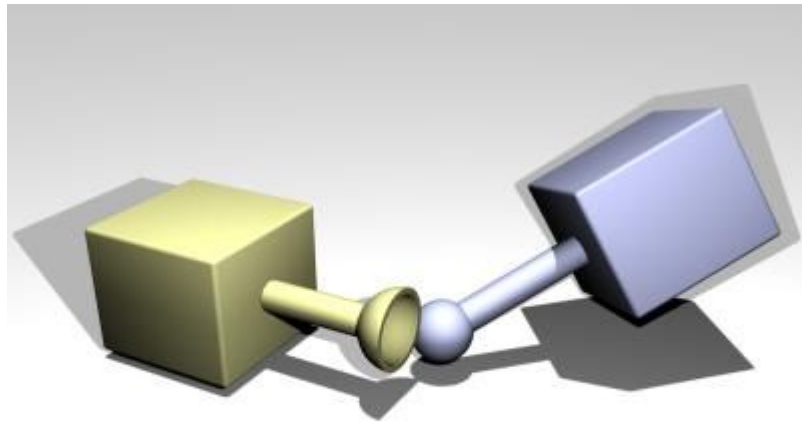


Ragdoll model in *Juice*

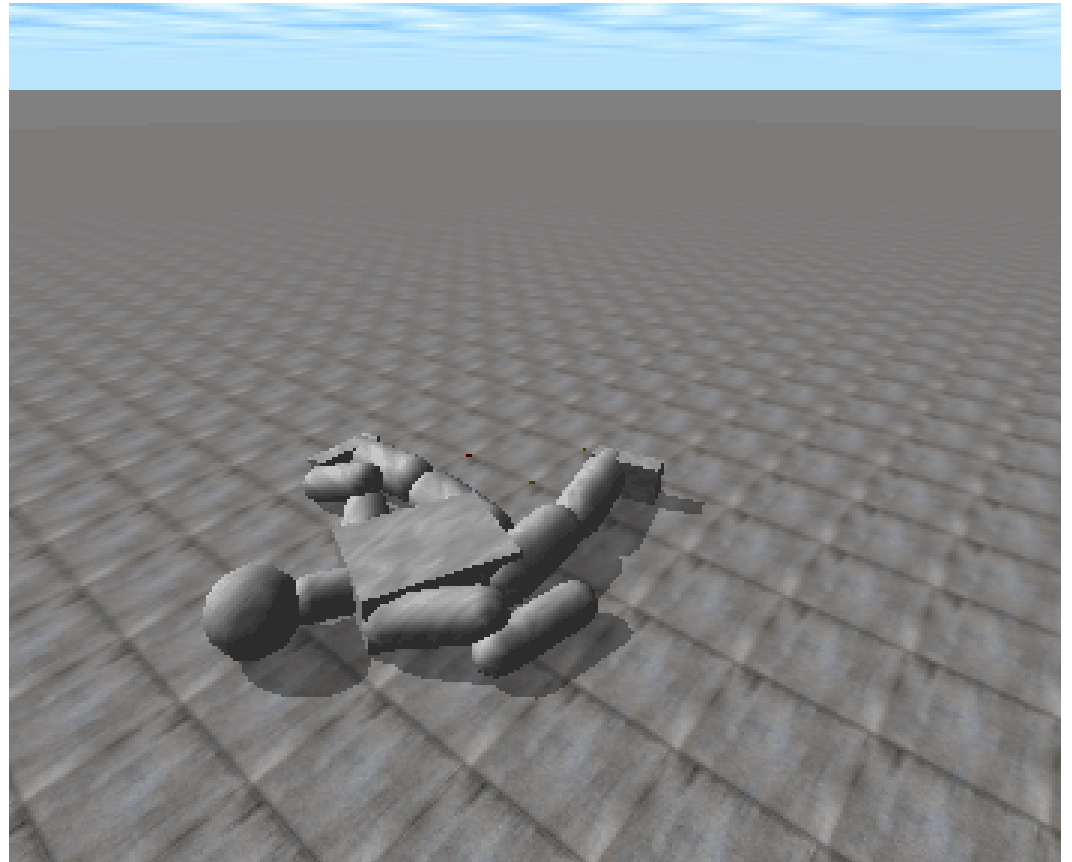
Case Study: *Stair Dismount* and Sequels



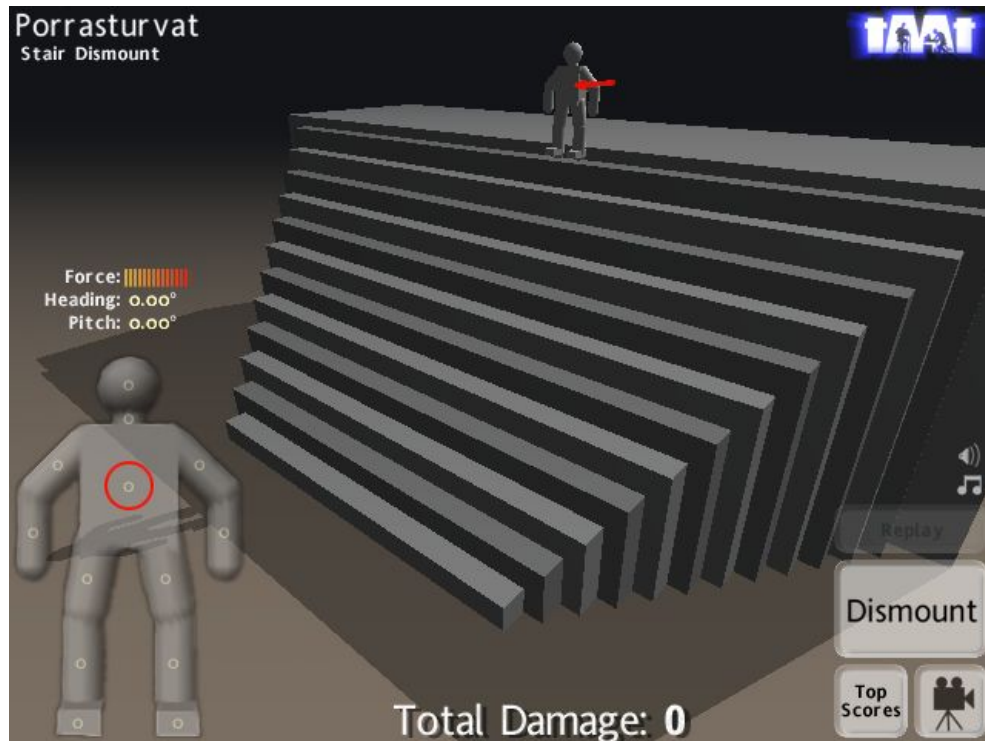
Case Study: *Stair Dismount* and Sequels



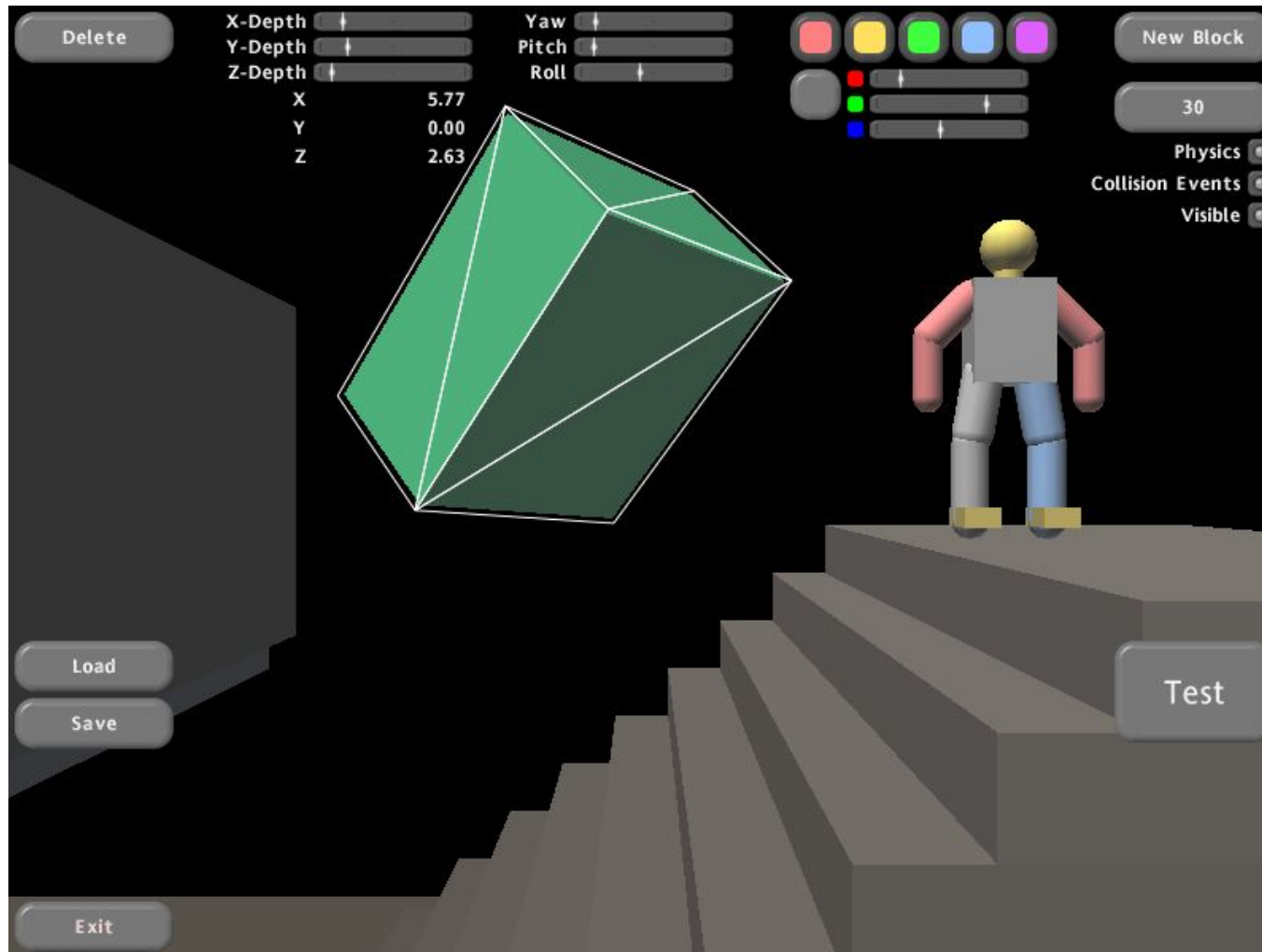
[Smi04]



Case Study: *Stair Dismount* and Sequels



Case Study: *Stair Dismount* and Sequels



Questions & Answers

- My home page: <http://jet.ro>
 - This presentation will be available there.
- Other links
 - ODE: <http://ode.org>
 - Dismount games: <http://jet.ro/dismount/>
 - Juice: <http://www.natew.com/juice/>

- Jak01 Jakobsen, T., *Advanced Character Physics*, Game Developers Conference, 2001.
<http://www.gdconf.com/archives/2001/> [2005-06-21]
http://www.gamasutra.com/resource_guide/20030121/jacobson_01.shtml [2005-06-21]
- McL03 McLaurin, M., *Outsourcing Reality: Integrating a Commercial Physics Engine*, 2003.
http://www.gamasutra.com/resource_guide/20030121/maclaurin_01.shtml [2005-06-21]
- Smi04 Smith, R., *Open Dynamics Engine User Guide*, 2001-2004.
<http://ode.org/ode-latest-userguide.html> [2005-06-21]