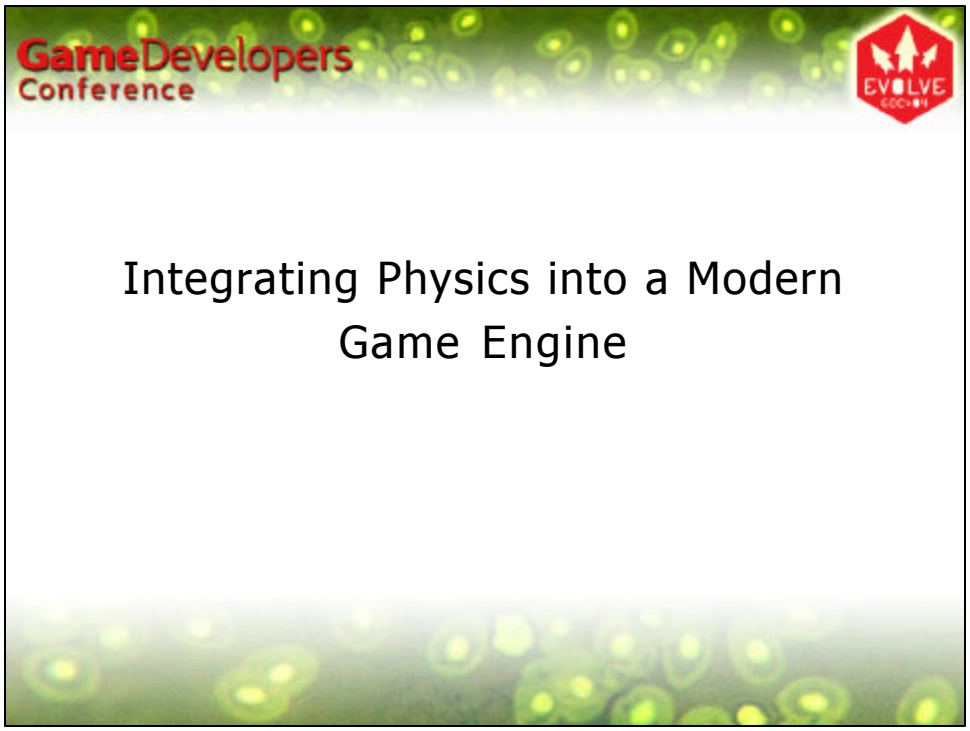
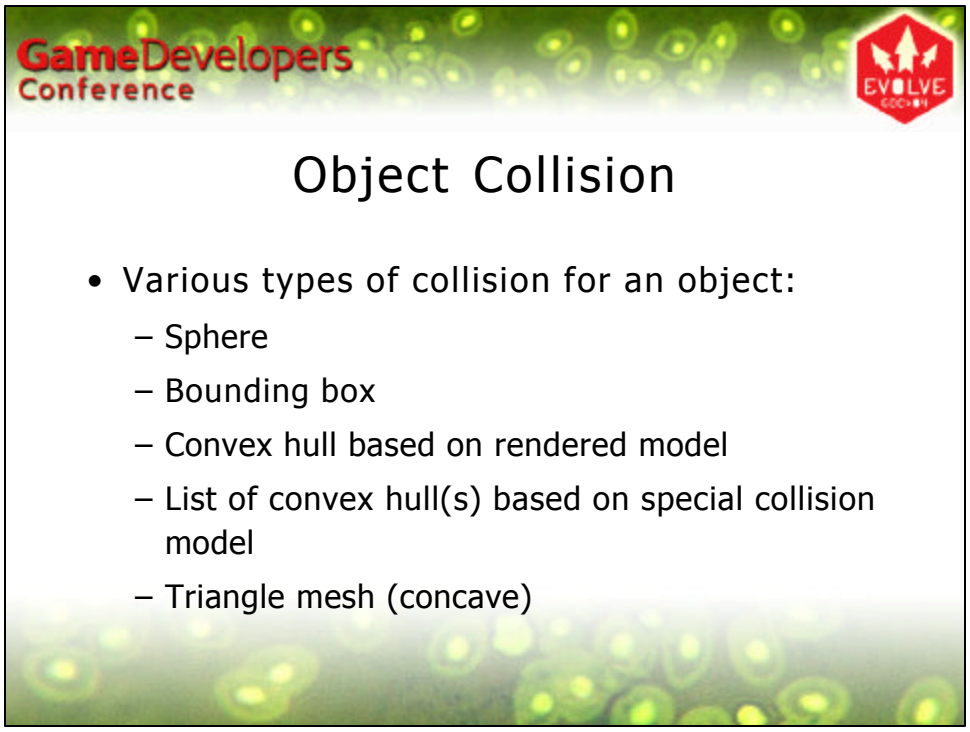

GameDevelopers
Conference
EVOLVE
Integrating Physics into a Modern
Game Engine


GameDevelopers
Conference
EVOLVE
Object Collision
• Various types of collision for an object:
– Sphere
– Bounding box
– Convex hull based on rendered model
– List of convex hull(s) based on special collision model
– Triangle mesh (concave)

GameDevelopers Conference

EVOLVE

Tools

- Fixed objects/World geometry
  - Collision for fixed objects collected during level export and appended to level file
- Mobile objects
  - Game tries to load pre-cached collision information, creates collision for any objects not in cache

GameDevelopers Conference

EVOLVE

Collision Cache

- Load-time object creation
  - Try to find shape in cache based on key information (type enum, template name, model name, etc)
  - If that fails, it creates new collision for the object and adds it to the cache
- Game can be run with a command-line option to load a particular level and write out the collision cache

GameDevelopers Conference

EVOLVE

Collision Cache

- Important
  - Drastically speeds up load times
  - Reduce memory requirements and fragmentation
  - Can reduce/remove some model information in certain cases
- Also allows for easy collision shape re-use

GameDevelopers Conference

EVOLVE

Raycasts

- We wrote our own raycast system that queries collision on a per-object basis. This allows us to:
  - Only test against objects based on the ray's traversal of our scene graph
  - Allows for very customized filtering of objects (based on object type, flags, etc)

GameDevelopers Conference

EVOLVE

Collision Filtering

- Very important to allow a variety of different collision options
- Each object belongs to one collision group. For example:
  - 'Debris' collides only with the 'world' collision group. Used for gibs, small broken pieces, etc.
  - 'Tracker' collides with most everything but 'world'. This is used for world objects that move (doors, etc).

GameDevelopers Conference

EVOLVE

Collision Filtering

- Collision filter also works based on object flags (collision temporarily disabled, etc)
- Can prevent 2 objects from colliding with one another temporarily (list of pairs). Example: While the player is on a ladder, collision between the player and the ladder is disabled

GameDevelopers Conference

EVOLVE

Collision Events

- Many things can be done automatically from handling collision events:
  - Damage
    - Ignore collision if damage kills/breaks object
  - Sounds
    - Modify volume based on impact forces
    - Notify nearby AI so they can react (only if the result of a player action)
  - Begin/end character interactions
    - Player touches ladder, pole, AI, etc

GameDevelopers Conference

EVOLVE

Collision Events

Demo

GameDevelopers Conference

EVOLVE

Player Collision

- Must be smooth to not catch on corners
- Bottom should be sloped for moving up stairs/ramps

We used 2 spheres to represent the body, and a cone-shaped hull for the 'foot'.

GameDevelopers Conference

EVOLVE

Player Collision

- Generally for a bipedal character it is better to prevent the collision from rotating (don't want the player to turn, wobble, or fall over based on collision).
  - Best results came from setting an inertial tensor to prohibit rotation.

GameDevelopers Conference

EVOLVE

Moving the Player / AI

- Multiple approaches:
  - Modify position directly
  - Set velocity
  - Apply impulses

GameDevelopers Conference

EVOLVE

Moving the Player / AI

- Character movement types
  - Controller-based
  - Animation-based
  - Combination

GameDevelopers Conference

EVOLVE

Moving the Player / AI

- Controller-based
  - New Velocity = [Old Velocity] + (Player input / AI goals, etc)
    - Example: Walk, run

GameDevelopers Conference

EVOLVE

Moving the Player / AI

- Animation-based
  - Velocity is based on movement of the root bone each frame. Allows for very specific movement.
    - Example: Attacks, hit reactions that move the character
  - "Uber root"
    - Important to move the collision with or slightly in front of the model. We allowed full artist control over this by parenting the model's root joint to another joint that specifies the collision's movement.

GameDevelopers Conference

EVOLVE

Moving the Player / AI

- Combination (Animation + Controller)
  - Up velocity from the animation, forward/right from controller
    - Example: Jump + Air Control

GameDevelopers Conference

EVOLVE

Phantoms

- Shapes used for querying for interpenetrations
- Useful when checking for room to perform an action
  - Can the player stand?
  - Is there room for the player to allow pulling up from a ledge?
  - Can the AI jump here?

Constraint System

- Model structure
  - Array of Groups (collection of triangles sharing a shader)
  - Each group has a 'type' (byte) for referencing multiple related groups
- Shader system
  - Typically defines a unique visual effect, but very flexible
  - Parameters specified in a text file
  - Created special shaders to represent each constraint type (hinge, point to point, etc)

GameDevelopers Conference
EVOLVE

Constraint System

- Example shader:

```
example
{
  shader   blend
  texture  tex1
  texture2 tex2
  alpha    0.5
}
```

```
exampleConstraint
{
  shader   constraint_hinge
  parent   exampleParent
  bounds   (-45, 45)
  mass     20
  damp     0.75 0.25
}
```

GameDevelopers Conference

EVOLVE

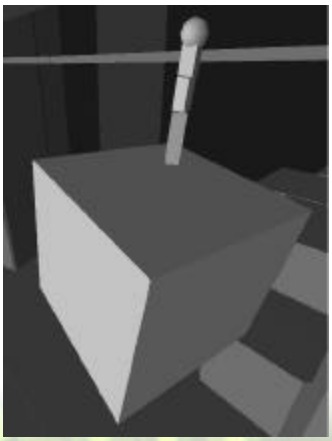
Constraint System

- Benefits of using model structure
  - Tool path already exists
  - Artists familiar with tools
  - Allows multiple bone skinning between constrained objects for improved visuals (rope, etc)

GameDevelopers Conference

EVOLVE

Constraint System



GameDevelopers Conference

EVOLVE

Constraint System

- Constraint Setup
  - Rigidbody and constraint created for each model group based on parameters in shader
- System is updated with model's skeleton
  - Anchors get their position from the skeleton
  - Others override their associated joint's position

GameDevelopers Conference

EVOLVE

Constraint System

- Attachment to characters
  - Create a small sphere rigidbody and add it to your constraint system as the root
  - During update, move this sphere with the attachment point on the model

GameDevelopers Conference

EVOLVE

Constraint System

- Attachment sequencing problem
  - If you update your constraint system at the same time as the object it is attached to, your anchor will always lag by a frame.
  - This generally isn't very noticeable unless the object the constraint system is attached to moves quickly
- Solution: When this problem isn't acceptable, you can fix this by adding the constraint system to a separate physics simulation, updating the anchor's position between updating the primary physics simulation and the constraint physics simulation.

GameDevelopers Conference

EVOLVE

Constraint System Example

- Hanging crate
  - Rope consists of multiple rigidbodies connected by point to point constraints
  - Rope connects to crate rigidbody also with a point to point constraint
  - Mark rope rigidbodies as breakable

If setup well, you can shoot the rope breaking a point to point constraint with automatic damage and sound when the crate lands.

GameDevelopers Conference

EVOLVE

Constraint System Example

Demo

GameDevelopers Conference

EVOLVE

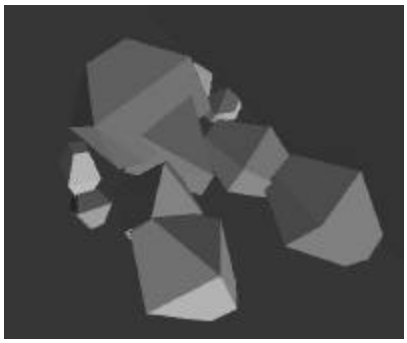
Ragdoll

- Extension of constraint system
  - Differences:
    - Overrides entire skeleton update
    - Setup re-positions to previous pose and calculates initial per-object velocity based on projected future pose
    - Object's position is set to 'main' rigidbody's center of mass during update (for scene graph)

GameDevelopers Conference

EVOLVE

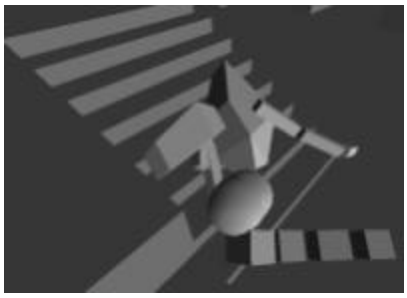
Ragdoll



GameDevelopers Conference

EVOLVE

Ragdoll



GameDevelopers Conference

EVOLVE

Ragdoll Attachment

- Multiple markers dictate attachment points on ragdoll
- Create constraint between closest attachment marker and a new 'anchor' rigidbody
- During update, move anchor rigidbody to attacher's position
- Can be used from anything from player or AI grabbing a ragdoll body, to attaching the ragdoll to some moving part of the world (meat hook, etc)

GameDevelopers Conference

EVOLVE

Breakable Objects

- Setup using model structure
- Either based on current model, or can swap models
- Object created for each model group type
- Original object removed
- Impulse based on cause applied

GameDevelopers Conference

EVOLVE

Breakable Objects




GameDevelopers Conference

EVOLVE

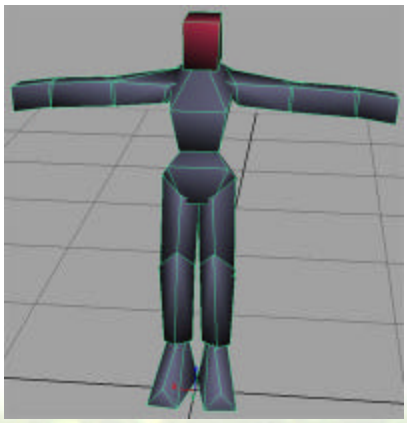
Hit Regions

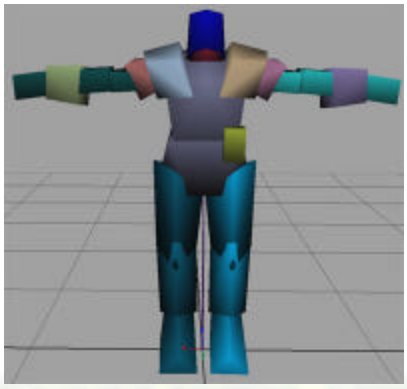
- Setup using model structure
- Convex hull generated for each model group
- Model group shader names resolve to Hit Region types enum (body, critical, limb, user0..9)
- During raycast, if ray passes quick-out checks, hulls are updated based on current skeleton, then raycast against.

GameDevelopers Conference

EVOLVE

Hit Regions





GameDevelopers Conference

EVOLVE

Shooting Parts off Characters

- Similar to Breakable Object setup
- Each Hit Region type has an associated model group type
- New object is created with convex hull based on verts of all model groups of that type
- Impulse based on cause applied
- Disable render of original object's model groups of that type

GameDevelopers Conference

EVOLVE

Shooting Parts off Characters

GameDevelopers Conference

EVOLVE

Explosions

1. Search for all objects based on explosion radius
2. Apply damage
3. Re-Search for objects based on explosion radius (previous objects may have broken into new objects, etc)
4. Apply blast force

GameDevelopers Conference

EVOLVE

Crushing

- Difficult problem to solve generally
  - Many potential situations
    - Closing door and wall
    - 2 doors sliding together
    - And much worse…
- Ended up going with specific, situational detection
- Could try detection based on interpenetration over multiple frames, interpenetration depth, etc.

GameDevelopers Conference

EVOLVE

Sequencing

- Important for an object's position to remain constant throughout a frame
- When going with a velocity/impulse model, sequence should be:
  1. Setup velocity changes (player/AI update)
  2. Render
  3. Step physics simulation
  4. Update object positions


GameDevelopers
Conference
EVOLVE


GameDevelopers
Conference
EVOLVE