Rigblocks: Player-Deformable Objects

Lydia Choy, Ryan Ingram, Ocean Quigley, Brian Sharp, Andrew Willmott

Maxis, Electronic Arts
Spore: Recap

• Want players to be able to create key parts of their game

• Pollinate player-created things via servers, so your game is made of both your own creations and others’

• Richer experience, less art work(!)
Spore: Recap

- Players create game assets
- Creatures, Buildings, Vehicles...
How can players create models?

• Let player use supplied parts to build model
  – Allow stacking, pinning, sliding

• *But,* static is boring, requires many blocks to be expressive. So
  – Add animations that *deform* blocks
  – Animations driven by player-controlled handles

• Result: Rigblocks, our LEGO\textsuperscript{tm}(R)(whatever)
Advantages

- Player interaction with the block is intuitive and straightforward
- Rigblock deformations are expressive
- Provides a balance between enabling player creativity and amplifying player creativity
Advantages

• Aiming for the sweet spot between:
  
  – High-quality, artist-created models, with no player control
  
  – Lower-quality, effort-intensive, wholly player-driven approach, such as providing a sculpting tool.
Animation Deforms Mesh
Animation Deforms Mesh

DeformAxisForward

DeformAxisUp

DeformAxisLeft
Animation Deforms Mesh
Note: Creatures

• Base block is a special block: body mesh

• Allow player control over a basic skeleton
  – Adjust spline, glue limbs

• Mesh generated via metaballs

• Rigblocks attached to body
Storyboarding

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="be_fortress_01" /></td>
<td><img src="image2" alt="be_fortress_02" /></td>
<td><img src="image3" alt="be_fortress_03" /></td>
<td><img src="image4" alt="be_fortress_04" /></td>
</tr>
<tr>
<td><img src="image5" alt="be_fortress_05" /></td>
<td><img src="image6" alt="be_fortress_06" /></td>
<td><img src="image7" alt="be_fortress_07" /></td>
<td><img src="image8" alt="be_fortress_08" /></td>
</tr>
<tr>
<td><img src="image9" alt="be_fortress_09" /></td>
<td><img src="image10" alt="be_fortress_10" /></td>
<td><img src="image11" alt="be_fortress_11" /></td>
<td><img src="image12" alt="be_fortress_12" /></td>
</tr>
<tr>
<td><img src="image13" alt="be_fortress_13" /></td>
<td><img src="image14" alt="be_fortress_14" /></td>
<td><img src="image15" alt="be_fortress_15" /></td>
<td><img src="image16" alt="be_fortress_16" /></td>
</tr>
</tbody>
</table>
Storyboarding
Storyboarding: A Single Block
Pipeline

• Standard workflow: separate author file per animation

• Rigblocks: Multiple animations, so use track editor

• MEL scripts control addition of handle rigs
  – Handles drive animation! (Via expressions)
  – Artist places handle, so can iterate in-Maya
Animation Technology

• Can’t use standard animation blending

\[ 50\% \text{ Def}_A + 50\% \text{ Def}_B \neq \text{Average}(A, B) \]

• Use cumulative blending from rest pose
  – Match Maya by composing deform matrix at end
    from separately accumulate scale, rotate, translate

• Multiblender
  – Handles standard “runtime” animations
  – Applies deforms on top
Baking

• Remove all deform animations, producing a new base mesh

• Model must be able to be rendered at game rates
  – Single texture page, single material
  – Generate LODs
Baking: Animation

• Desirable for blocks to carry “runtime” animations through (e.g. mouths)

• But such rigblocks must be substituted with low-bone-count versions

• Requires retargetting composite deform pose to new runtime skeleton (base pose has changed)
Runtime Animation

Authored Block
Runtime Animation

- Many bones
- Skeletal animation
- Blendshape animation

- Reduced skeleton
- Skeletal animation
Runtime Animation

- Apply deformation handle

- Mesh is retargeted to new (runtime) skeleton
Runtime Animation

Runtime Block

- Runtime animations are retargeted to new skeleton
Thanks

• All the Rigblock artists
  – Umaru Jalloh, Mike Khoury, Ferby Miguel, Jane Ng, Holly Ruark, Matt Small

• The Editor team
  – Dave Culbya, Chaim Gingold, Alex Lam, Dan Moskowitz
Questions?