Course Plan

- lec. 1: Introduction
- lec. 2: Mathematics for 3D Games
- lec. 3: Scene Graph
- lec. 4: Game 3D Physics +
- lec. 5: Game Particle Systems
- lec. 6: Game 3D Models
- lec. 7: Game Textures
- lec. 9: Game Materials
- lec. 8: Game 3D Animations
- lec. 10: 3D Audio for 3D Games
- lec. 11: Networking for 3D Games
- lec. 12: Rendering Techniques for 3D Games
- lec. 13: Artificial Intelligence for 3D Games

computer animation

162

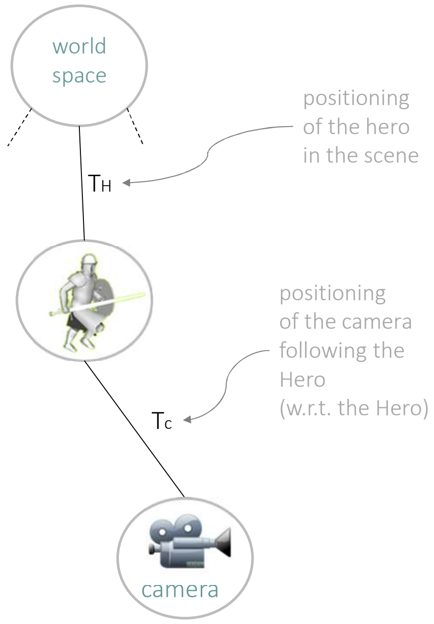
Integration of skeletons in the scene-graph

- A skeleton (rig) can be considered just a subtree in the scene-graph
  - only, its local transforms are defined by the current frame in the current skeletal animation asset
- Examples of common reasons to use the skeleton just as a part of the scene graph:
  - Placement of the camera in a bone (e.g. the head bone)
  - Defining geometry proxies (hit-boxes) for collision detection in bones

163

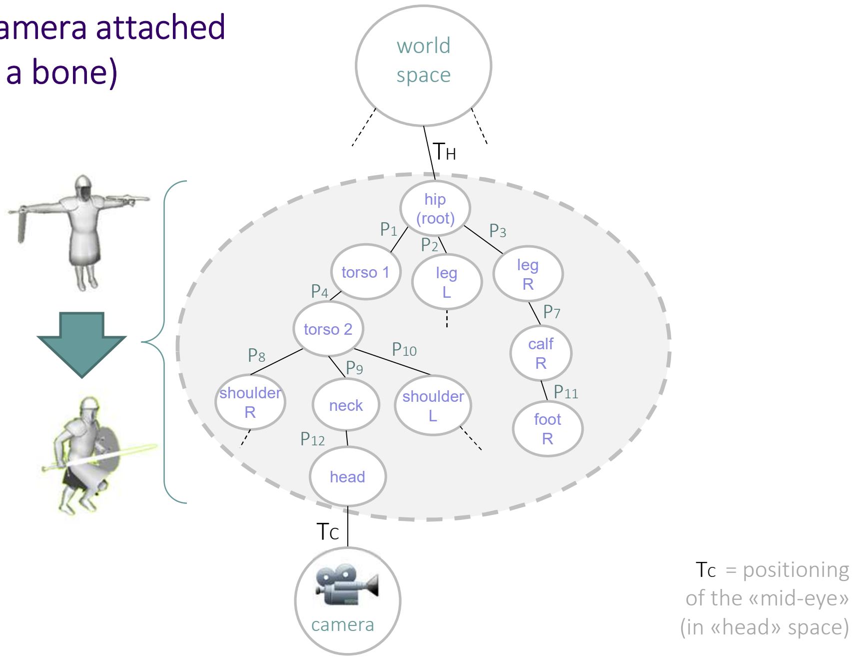
Marco Tarini
Università degli studi di Milano

Third person view



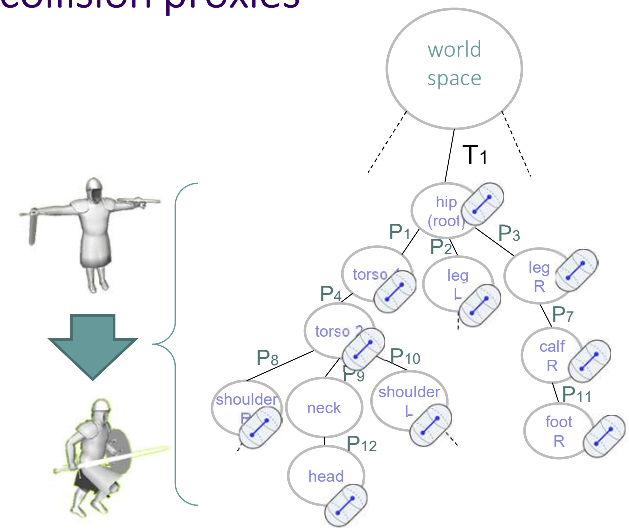
164

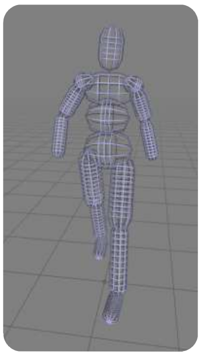
First person view
(camera attached to a bone)



165

Per-bone collision proxies





Hit-boxes: e.g., capsules
(not necessarily in *every* bone)

166

(Pre)processing tasks for skeletal animations: examples

- Keyframe sparsification
  - input: animation with N keyframes
  - output: animation with M<N keyframes
- Animation Retargeting
  - input: SkelA + Animation for SkelA + Skel2
  - output: a similar Animation, but for Skel2
- Automatic generation, from a blend-shape
  - input: a blend-shape
  - output: Skeleton + Skinned Mesh + Anim
  - note: the opposite is a trivial (it's a form of *baking*)

167

Sparsification of keyframes (reduce number of keyframes)

- Objective: removal of redundant keyframes
  - "Redundant" = can be approximated by in-betweens
  - A preprocessing task
- Basic algorithm concept:
  - for each keyframe P_x
    - tentatively remove P_x
    - compute interpolated version P_i from remaining keyframes
      - the prev. and next ones
    - if *distance*(P_i , P_x) > MAX_ERR then reinsert keyframe P_x

168

Ragdolling (notes)

- Idea: let a physical simulation determine the evolution of the skeleton (and attached geom. proxies)
  - Includes: gravity, external forces, collisions with other objects, self-collisions (i.e., collision between proxies associated to the bones)
- Ingredients:
  - Per-bone proxies (in at least a subset of the bones)
  - Constraints, such as… attachments of bones, constraint on rotations (e.g., "knees don't bend backward or sideways")
  - The latter can be expressed as positional constraint in a Position Based Dynamics simulation
- Result: procedural skeletal animation!

169

Some of the benefit of skinning (recap)

- Animations and animated models are reciprocally orthogonal
  - Can be produced / stored independently of each other
  - 100 animations for 100 models: 100+100 assets, *NOT* 100x100 assets
  - That is: it's easy to retarget animations over different models
- Models appear deformed by pose on screen,
  but only rest models and animations need be stored in VRAM
- Tremendous interpolation power
  - Thanks to Forward Kinematics
  - Keyframes can be very far and spare
    and still produce good in-betweens
- Flexible: possible to dynamically combine different animations into new ones
  - By animations interpolation, or by layering
- Efficacious ways to capture, edit, compute, simulate anims
  - Leveraging IK, physics simulations

Important:
"local" transforms are manipulated.
Them we need to combine them into "final" transform

170

Some limitations of skinning (recap)

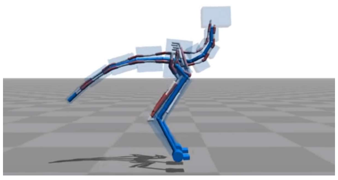
- **Deformations** (induced on the skinned mesh by the pose)
  are simple and may be unrealistic
  - Nothing of: dynamic effects, collisions (contact surfaces), volume preservation…
  - This can in part be helped by:
    good skinning (by skillful 3D artist)
    additional bones
  - DQS is arguably a bit better than LBS, but marginally
- Interpolated / layered / IK-ed / rag-dolled **animations** can be unrealistic / simplistic / unaware of the broader context

In summary, (blend) skinning is an extremely cost-effective technique, providing a wealth of complex effects for a small cost
(in terms of computation, memory, construction times…),
but the quality of animations is intrinsically limited and may become a bottleneck games visual-quality in the next future.

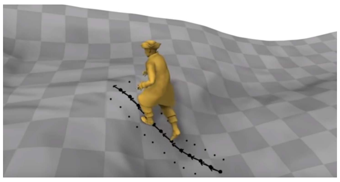
171

Research topic: apply ML to generate skeletal animations

- A very active area of research…



Flexible Muscle-Based Locomotion for Bipedal Creatures
Thomas Geijtenbeek, Michiel van de Panne, A. Frank van der Stappen
SIGGRAPH 2013



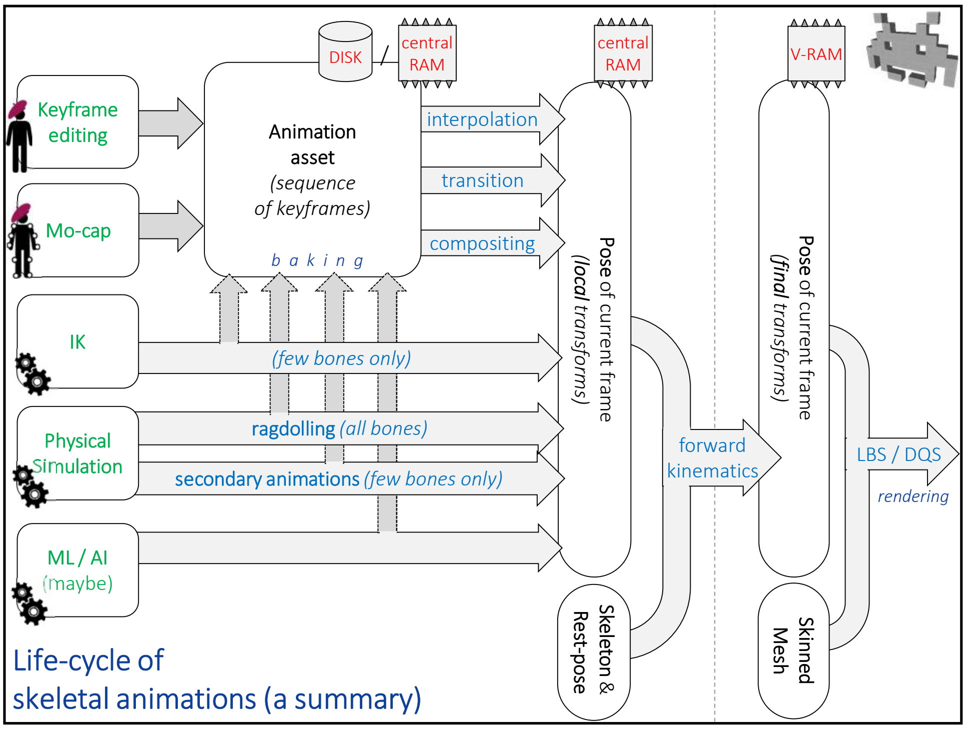
Phase-Functioned Neural Networks for Character Control
Daniel Holden, Taku Komara, Jun Saito
SIGGRAPH 2017

(among MANY others)

Next lecture!

172

Life-cycle of skeletal animations (a summary)



174

Observation: Blend-shapes & Skeletal Animations blend well with...

- Texturing!
  - UV coords are only defined in rest shape
  - they are shared by all frames
  - Textures are shared by all frames
- Micro-meshes!
  - Blend-shapes:
    Base mesh and displacement directions are both defined per morph-target
  - Skeletal animations:
    skinning is defined on base-mesh only,
    deforms both vertex positions and displacement dirs.
  - *Both:* micro-displacem. are applied on top of animated mesh

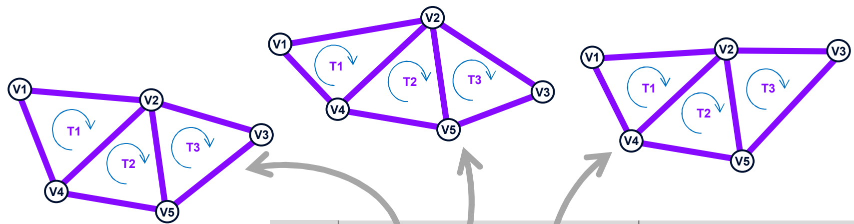
175

Observation: blandshapes + skeletal anims *can* be used togheter

- A blend shape can be skinned!
- Both animations can be combined
  - frame of the blend shape
    $\neq$
    frame of the skeletal animation
- Examples:
  - Breathing animations = blend shape,
    + Idle animation = skeletal anim
  - Cheeks puffing = blend shape (face morph)
    + mandible bone = skeletal animations
  - Blend shapes correctives (see later)

176

Observation: blandshapes + skeletal anims *can* be used togheter



Connectivity (shared):			
Tri:	Wedge 1:	Wedge 2:	Wedge 3:
T1	4	1	2
T2	4	2	5
T3	5	2	3

Vert:	*geometries:*				*shared attributes:*		
	Base Shape	Shape 1	Shape 2	Shape …	UV	Bone links (skinning)	…
V1	(x,y,z)	(x,y,z)	(x,y,z)	…	(u,v)	$(b_0,w_0,b_1,w_1,b_2,w_2)$	
V2	(x,y,z)	(x,y,z)	(x,y,z)	…	(u,v)	$(b_0,w_0,b_1,w_1,b_2,w_2)$	
V3	(x,y,z)	(x,y,z)	(x,y,z)	…	(u,v)	$(b_0,w_0,b_1,w_1,b_2,w_2)$	
V4	(x,y,z)	(x,y,z)	(x,y,z)	…	(u,v)	$(b_0,w_0,b_1,w_1,b_2,w_2)$	
V5	(x,y,z)	(x,y,z)	(x,y,z)	…	(u,v)	$(b_0,w_0,b_1,w_1,b_2,w_2)$	

177

Rendering Skinning + Blendshapes

To render a mesh…

- Load…
  - make sure all data is ready in GPU RAM (including skinning weights)
    - Geometries + Attributes
    - Connectivity

    THE MESH ASSET

    - Pose
      final transforms per bone

    THE ANIMATION (1 FRAME)

    - Textures
    - Shaders
    - Material Parameters…

    THE MATERIAL ASSET
- …and Fire!
  - issue the Draw Call

178

Example: Skinning with LBS + Blend-shapes with relative encoding

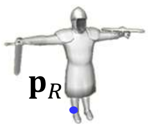
Per-bone final transform

Vertex position in rest pose

$$\mathbf{p}_P = \left(\sum_{i=0}^{3} w_i \, \mathrm{T}[b_i]\right)(\mathbf{p}_R + \vec{d})$$

bone links (mesh skinning)

relative blend-shape
Note: it is be applied in rest-pose, *before* the skinning



rest pose

deformed mesh

179

Limits of skinning (both LBS and DQS) Notes:

The bar for 3D game quality has gone up, but skeletal animations + skinning stayed the same for 10+ years.

Problems with deformations of the rest pose mesh:

- Does not account for Dynamic effects: e.g., a fat belly jiggling up and down during a run
  - Solution 1: use blend shapes (e.g., blend shape correctives)
  - Solution 2: add new bones (belly bone), add a dynamic simulation to control the bones (aka to control the "secondary motions")
- Does not account for collision/contacts
- Does not account for volume preservation
  - E.g., no muscle bulging
  - Can be in part compensated with skillful edit of bone weights

180

Limits of skeletal animations notes:

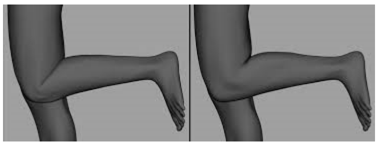
The bar for 3D game quality has gone up, but
skeletal animations + skinning stayed the same for almost 10 years.

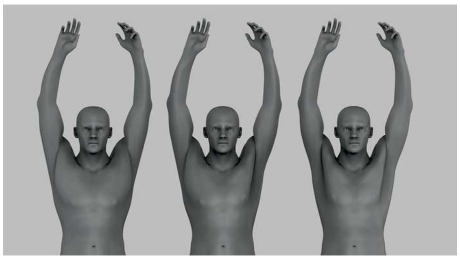
Problems with the skeletal animations:

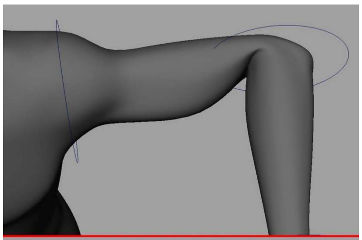
- Transitions between animation is crude, can look robotic
  - Possible solution: use ad-hoc animations for transitions
- Ragdolling = completely death "sack of potatoes"
  Authored Animation = character completely alive and in control, irresponsive to actual forces / dynamics
  - What about intermediate situations?
- IK not necessarily realistic
  - E.g.: feet are placed on the ground (not into it),
    but this is not how you would walk over a rugged terrain
- Animations are not physically based

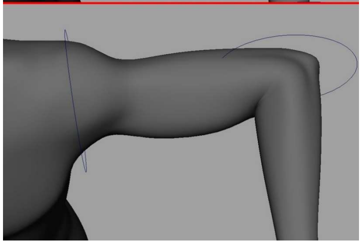
181

Blend Shape Correctives



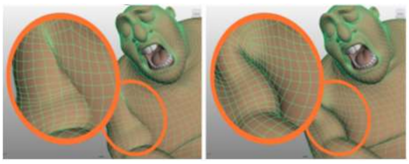






182

Research topic: Deformation beyond standard skinning

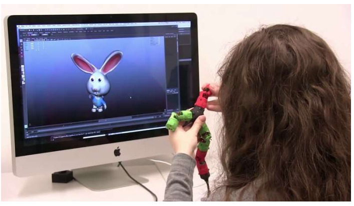


Efficient Elasticity for Character Skinning with Contact and Collisions
Aleka McAdams et al (Disney animation)
SIGGRAPH 11

Note: usually way more complex than direct methods (LBS / DQS).
More offline animation oriented than videogames

183

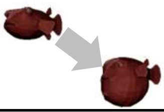
Research topic: better interfaces to author animations



Tangible and Modular Input Device for Character Articulation
Alec Jacobson, Daniele Panozzo, Oliver Glauser, Cedric Pradalier, Otmar Hilliges, Olga Sorkine-Hornung
SIGGRAPH 2014

184

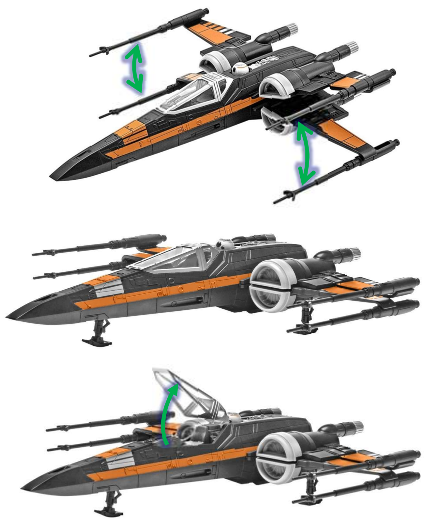
Asset animations in games

	Non-Procedural (ASSETS)	Procedural (e.g. PHYSIC ENGINE)
Rigid 	Kinematic animations	Rigid body dynamics
Articulated 	Skeletal Animations	Ragdolling; Inverse kinematics
Free form 	Blend-Shapes	~~(general) soft body simulation~~ *usually too expensive*; Cloth/ garments; Ropes

186

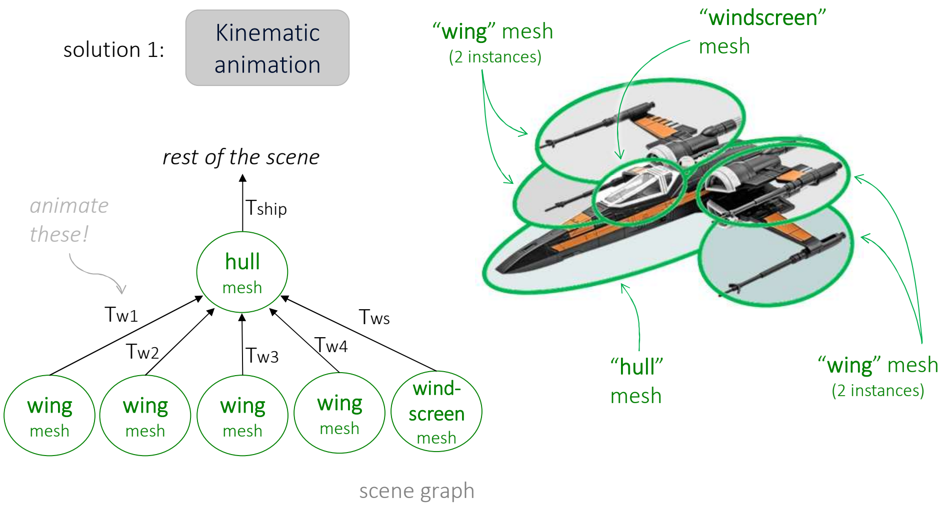
Representations for animations: which type to choose?

EXAMPLE:
say we want
a model capable of
doing this:



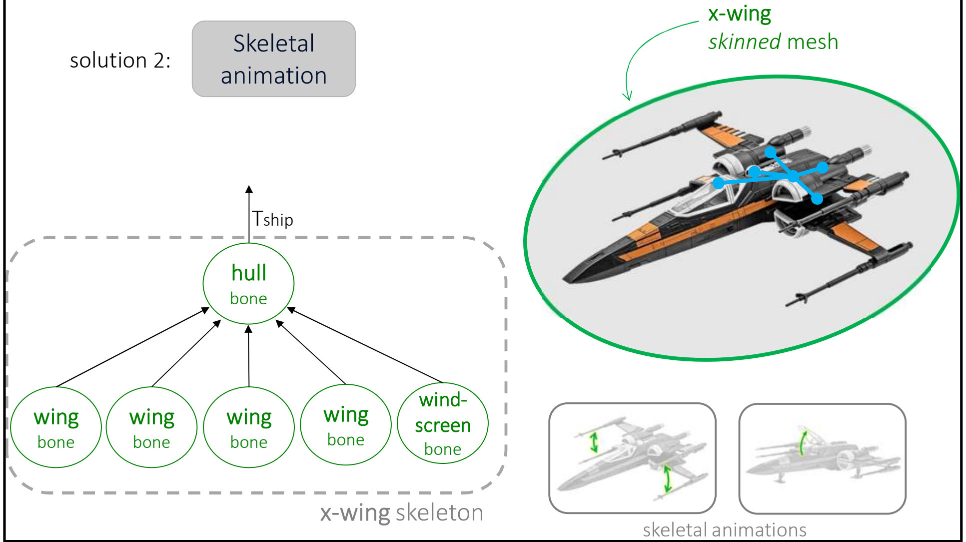
187

Representations for animations: which type to choose?



188

Representations for animations: which type to choose?



189

Representations for animations: which type to choose?

solution 3: Blend-shape

x-wing blend-shape

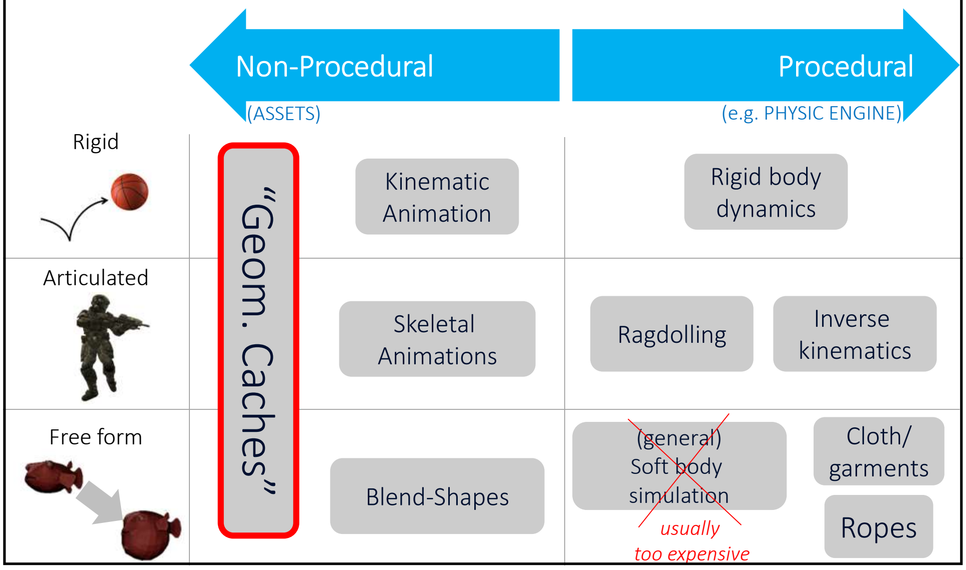
190

Representations for animations: which type to choose? In this example…

- Kinematic animation:
  - *how:* 3 (rigid) meshes, 5 instances, animate scene-graph transforms
  - can reuse geometry for all wings: most compact on RAM ☺
  - simpler rendering (just rigid meshes) ☺
  - 5 individual draw calls! ☹
  - limit: no deformable bits (e.g., no cloth pieces connecting the windscreen)
- Skeletal animation:
  - *how:* one skeleton + one skinned mesh + 2 skeletal animations
  - (in this case, single bone per-vertex is enough for the skinning)
  - open-windscreen with both open OR closed wings? Still possible! (layering of animations)
  - rendering: small increase of per-vertex computation burden: LBS or DQS ☹
  - single draw-call! ☺
  - deformable bits: allowed ☺
- Blend-shape:
  - *how:* one base-shape + 2 morphs
  - bad quality of interpolation: linear
    - vertices follow straight paths
  - heaviest on RAM ☹
    - not a problem, if very low-res
  - single draw call! ☺
    - to different buffers each frame

191

Animations in games
(of 3D Solid Objects)



193

Geometry Caches
(for lack of a better name)

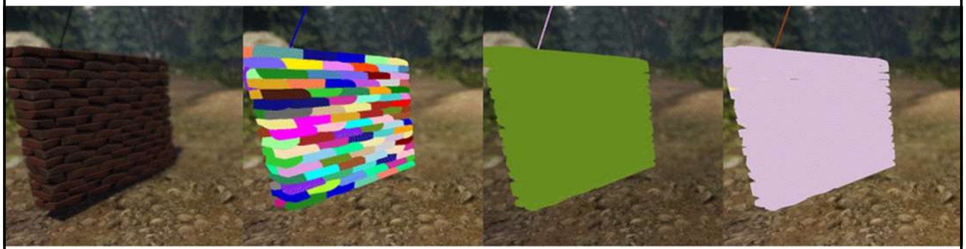
- Baked, optimized animations
  - of a mixture of types, like
    - blend shapes
    - kinematic animations
    - skinned animations
  - optimized
    - compressed, streamed…
  - Can be used to bake results of a physical simulation
    - i.e., convert it from procedural to kinematic

one used file format: .abc ALEMBIC by SONY PICTURES imageworks

194

Geom. Caches
(for lack of a better name)

- Baked, optimized animations
  - of the appropriate types including mixtures



Input:
170 Meshes
88400 Verts

as Pre-made Tansforms:
Meshes: 170
Data rate: 0.13 MB/s
Draw calls: 170
(same ones each frame)

as a Blend Shape:
Meshes: 1, with N shapes
Data rate: 4.3 MB/s
Draw calls: 1
(different one each frame)

as a Skeletal Animation
Meshes: 1, w skinning (*)
Data rate: 0.13 MB/s
Draw calls: 1
(same one each frame)
(*) just 1 bone per vertex

Geometry Caches
(a subset of Alembic)

by CRYENGINE

195

Animations in Mecanim (Unity) (notes)

- Assets (models, animation, skeletons) imported as formats:
  - fbx, collada
- Keyframe sparsification, or reduction of num of links per vertex
  - available during import / builds
- «Animator Controller» module → deals with:
  - blending between animations: «transitions»
  - compositing animations: «layers»
    - e.g.: a layer overwrites upper body bones
  - and is nicely WYSIWYG and has a nice graph GUI
- Inverse Kinematic: with scripts (`Avatar.SetIKPoistion`)
- Skeletons:
  - custom skeletons can be used
    (imported as assets)
  - OR, a standard built-in humanoid skeleton provided
    - ~21 bones
    - simplifies: rigging, ragdolling (predefined constrains), layers (predef. labelling)

196