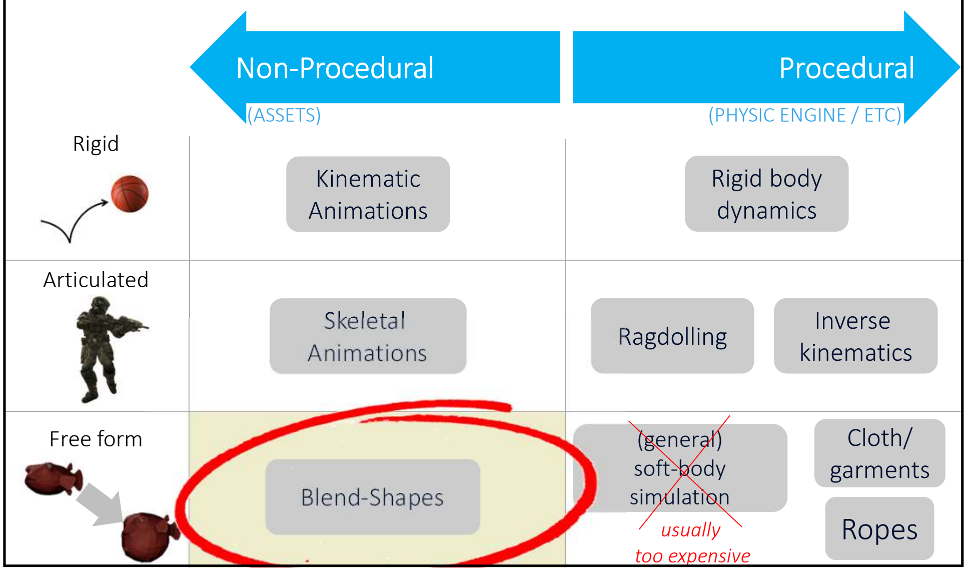
Course Plan

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: Networking for 3D Games ●
lec. 11: 3D Audio for 3D Games ●
lec. 12: Rendering Techniques for 3D Games ●
lec. 13: Artificial Intelligence for 3D Games ●

13

Animations in games
(of 3D Solid Objects)



Non-Procedural (ASSETS)

Procedural (PHYSIC ENGINE / ETC)

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

14

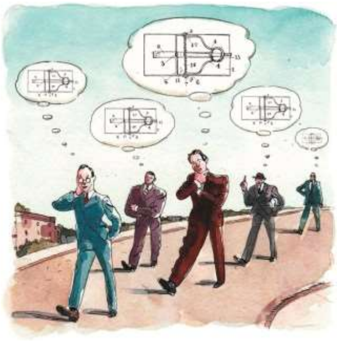
Animations in games: authored, procedural… or a mix?

- A few examples of current commonly used mixes:
  - 1: *"primary"* animations: authored
    "secondary" animations: physically generated
  - 2: *alive* characters: authored
    dead characters: physically generated ("ragdolls")
  - 3: walk cycle: authored (skeletal animation)
    exact *feet placement*: procedural (inverse kinematic)
  - 4: normal "behavior", such as sparring: authored
    gaze control during sparring: procedural
  - 5: normal "behaviors" such as jumping, running: authored
    modifications / transitions: AI generated
  - and more!
- mixing AI-generated with authored animations is a frontier in the field of Computer Animation!

15

Asset for free-form animations: Blend-shapes

- Also known as:
  - Morph-targets
  - Face-morphs
  - Shape-keys
  - Per-vertex animations
  - Vertex-animations
  - …



BARRY BLITT (THE NEW YORKER)

16

Blend shapes: concept

Walk cycle
(Monkey Island
LucasArt 1991)

- Animation in 2D (old school) games: a sequence of sprites
- Animation in 3D games: just a sequence of meshes?

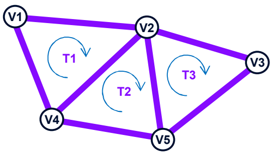
17

Reminder: representation of a mesh

- Indexed mode :
  - Geometry:
    - a 3D position for each vertex
  - Attributes:
    - more data, also stored in each vertex
    - (to be interpolated inside faces)
  - Connectivity:
    - Array of triangles (faces)
    - Each triangle = a triplet of indexes to vertex

18

Mesh (data structure)



connectivity (indexed)

Tri:	Wedge 1:	Wedge 2:	Wedge 3:
T1	V4	V1	V2
T2	V4	V2	V5
T3	V5	V2	V3

geometry:

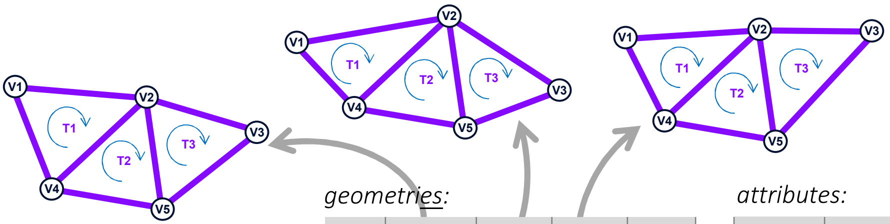
Vert:	Pos
V1	$(\mathbf{x},\mathbf{y},\mathbf{z})$
V2	$(\mathbf{x},\mathbf{y},\mathbf{z})$
V3	$(\mathbf{x},\mathbf{y},\mathbf{z})$
V4	$(\mathbf{x},\mathbf{y},\mathbf{z})$
V5	$(\mathbf{x},\mathbf{y},\mathbf{z})$

attributes:

UV	Col
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$

19

Blend shapes (data structure)



connectivity (indexed)

Tri:	Wedge 1:	Wedge 2:	Wedge 3:
T1	V4	V1	V2
T2	V4	V2	V5
T3	V5	V2	V3

geometries:

Vert:	Base Shape	Shape 1	Shape 2	…
V1	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	…
V2	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	…
V3	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	…
V4	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	…
V5	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	$(\mathbf{x},\mathbf{y},\mathbf{z})$	…

attributes:

UV	Col
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$
$(\mathbf{u},\mathbf{v})$	$(\mathbf{r},\mathbf{g},\mathbf{b})$

20

Blend shapes

- A mesh with several associated geometries
- I.e. a sequence of meshes (‘shapes’) with
  - shared connectivity
  - many shared attributies
    - except normals / tangents dirs
    - shared UV-map, per vertex colors…
  - different geometries
  - (and shared textures as well)
- Variants (they are equivalent):
  - Relative mode:
    - *base shape*: stored as per-vertex positions (points)
    - any other *shape*: stored as difference with *base shape* (vectors)
  - Absolute mode:
    - each *shape* stored as per-vertex positions (points)

aka ‘morph’
aka ‘morph-target’
aka (key)-‘frame’
aka ‘shape-key’

21

Blend shapes
(as a data structure, e.g. C++)

- Indexed mesh :

```
class Vertex {
  vec3 pos;
  rgb color;
  vec3 normal;
};

class Face{
   int vertexIndex[3];
};

class Mesh{
  vector<Vertex> vert; /* geom + attr */
  vector<Face> tris;   /* connectivity */
};
```

22

Blend shapes
(as a data structure, e.g. C++)

- Blend-shape :

```
class Vertex {
  vec3 pos [ N_SHAPES ] ;
  rgb color;
  vec3 normal [ N_SHAPES ] ;
};

class Face{
   int vertexIndex[3];
};

class Mesh{
  vector<Vertex> vert; /* geom + attr */
  vector<Face> tris;   /* connectivity */
};
```

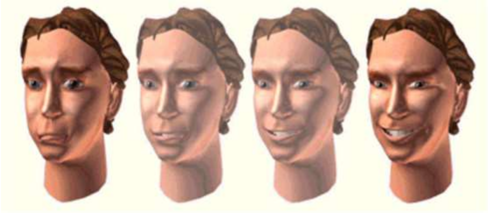
23

Blend-shapes: most common interchange formats

- Format supporting blend-shapes inlcude:
  - .GLTF (Khronos)
    "`morphTarget`", relative encoding
  - .DAE (Collada)
  - .FBX (Autodesk)
- Older / simpler alternatives:
  - .MD5 ("quake", valve)
  - or, just store a sequence of meshes (es .OBJ)
    - making sure connectivity is coherent!
      (vertex, face ordering must be the same – can be tricky)

24

Uses of Blend-Shapes: facial expressions



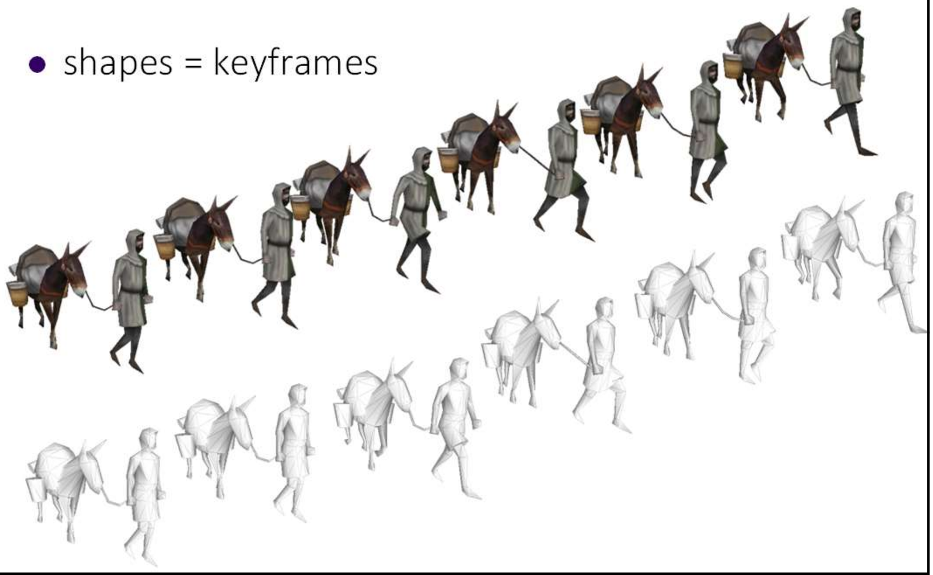
shape A

shape B

here: shapes = facial expressions
(typical use; that’s why they are also called “face morphs”)

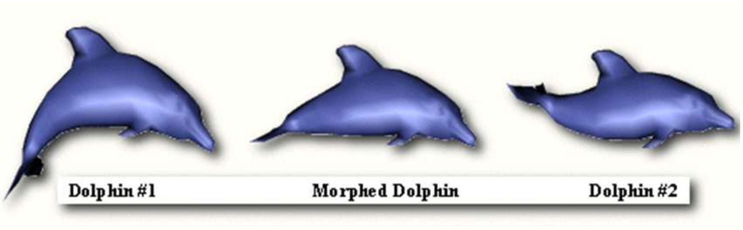
Uses of Blend shapes: temporal sequences

- shapes = keyframes



Uses of Blend shapes: temporal sequences

- Temporal sequences
  - shapes = keyframes

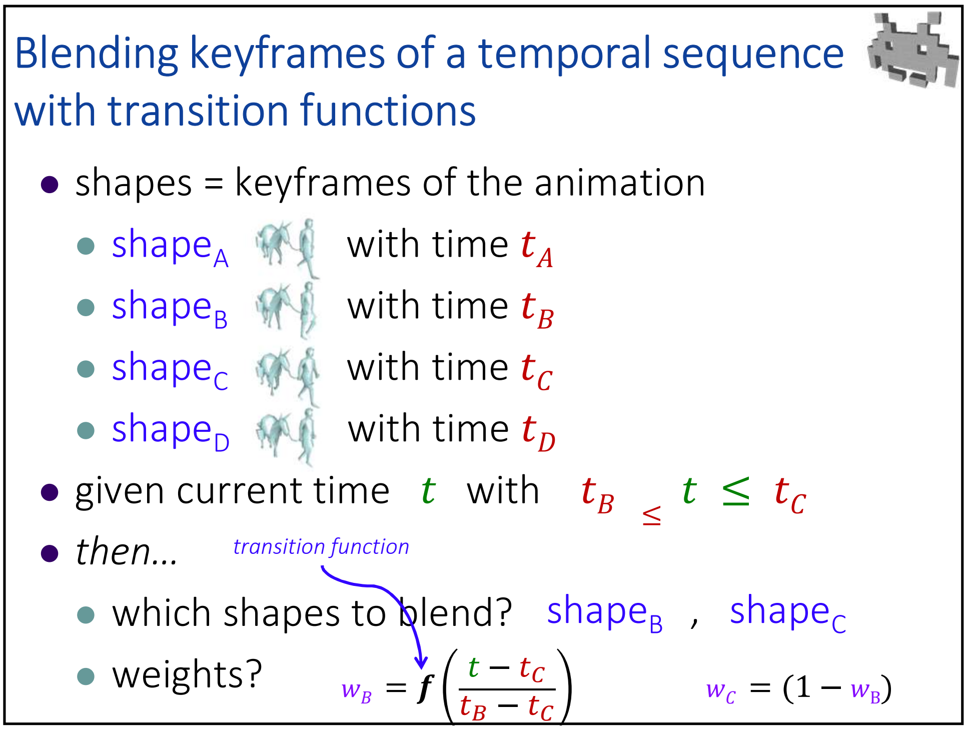


27

Blending keyframes of a temporal sequence

- shapes = keyframes of the animation
  - shape$_A$ with time t_A
  - shape$_B$ with time t_B
  - shape$_C$ with time t_C
  - shape$_D$ with time t_D
- given current time t with $t_B \leq t \leq t_C$
- *then…*
  - which shapes to blend? shape$_B$, shape$_C$
  - weights? $w_B = \dfrac{t - t_C}{t_B - t_C}$ $w_C = (1 - w_B) = \dfrac{t - t_B}{t_C - t_B}$

28



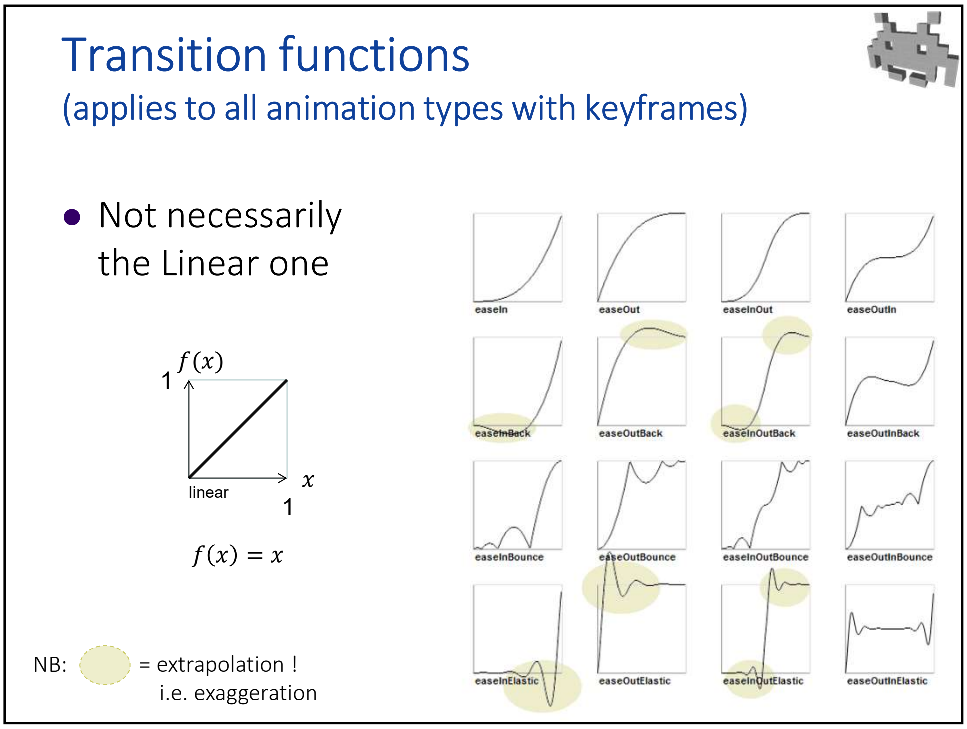
Blending keyframes of a temporal sequence with transition functions

- shapes = keyframes of the animation
  - shape_A with time t_A
  - shape_B with time t_B
  - shape_C with time t_C
  - shape_D with time t_D
- given current time t with $t_B \leq t \leq t_C$
- *then…*
  - which shapes to blend? shape_B , shape_C
  - weights?

transition function

$$w_B = \boldsymbol{f}\left(\frac{t - t_C}{t_B - t_C}\right) \qquad w_C = (1 - w_B)$$

29



Transition functions

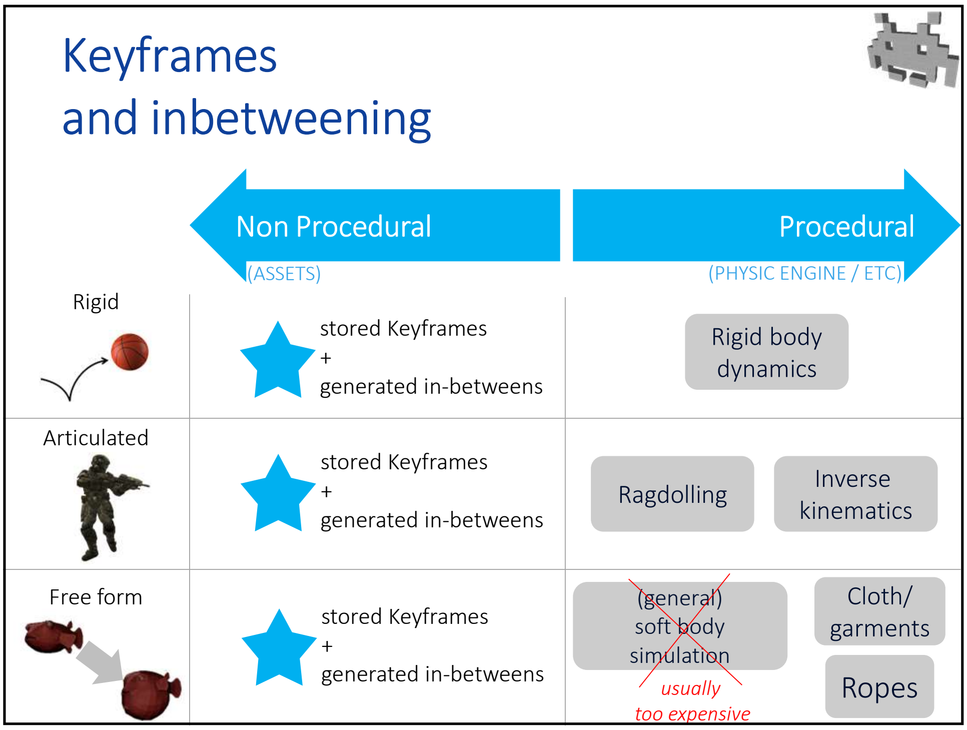
(applies to all animation types with keyframes)

- Not necessarily the Linear one

$f(x) = x$

NB: = extrapolation !
i.e. exaggeration

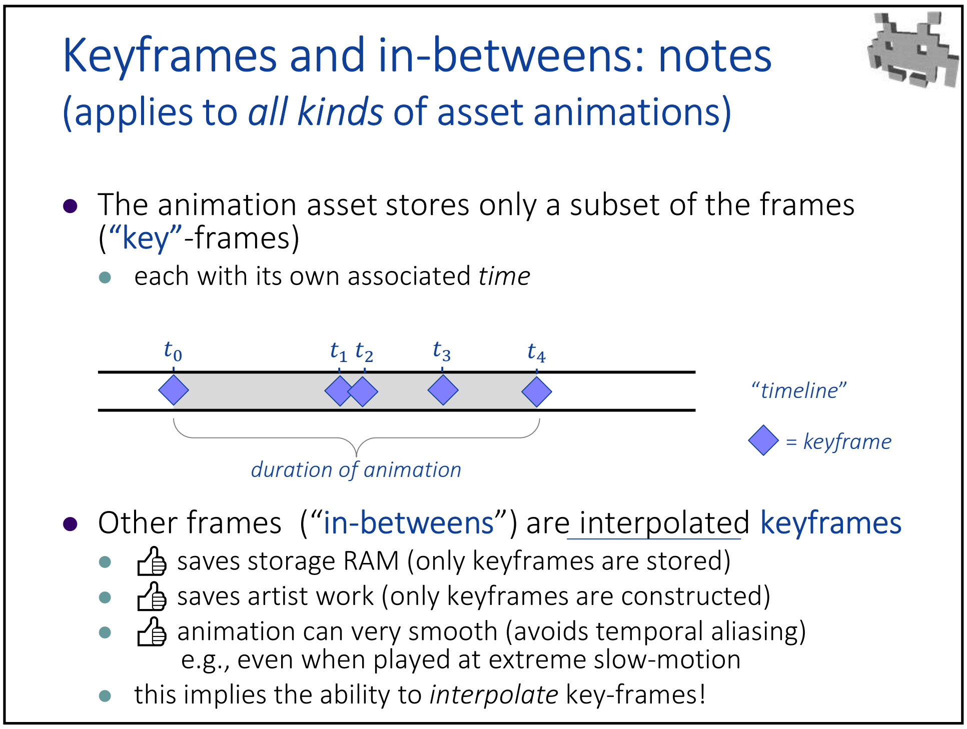
31



Keyframes and inbetweening

	Non Procedural (ASSETS)	Procedural (PHYSIC ENGINE / ETC)
Rigid	★ stored Keyframes + generated in-betweens	Rigid body dynamics
Articulated	★ stored Keyframes + generated in-betweens	Ragdolling Inverse kinematics
Free form	★ stored Keyframes + generated in-betweens	~~(general) soft body simulation~~ *usually too expensive* Cloth/ garments Ropes

32



Keyframes and in-betweens: notes (applies to *all kinds* of asset animations)

- The animation asset stores only a subset of the frames ("key"-frames)
  - each with its own associated *time*

- Other frames ("in-betweens") are interpolated keyframes
  - 👍 saves storage RAM (only keyframes are stored)
  - 👍 saves artist work (only keyframes are constructed)
  - 👍 animation can very smooth (avoids temporal aliasing) e.g., even when played at extreme slow-motion
  - this implies the ability to *interpolate* key-frames!

33

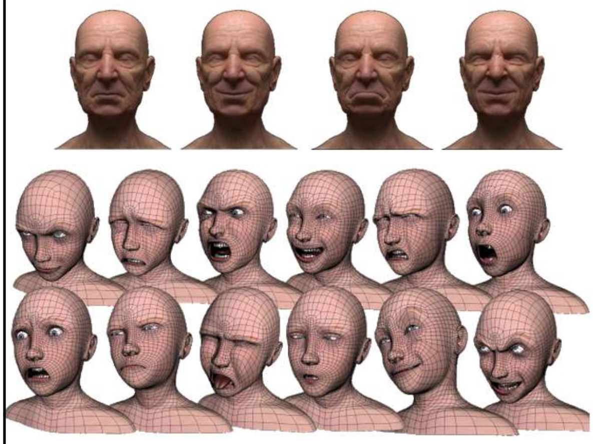
Keyframes and in-betweens: notes
(they apply to *all kinds* of asset animations)

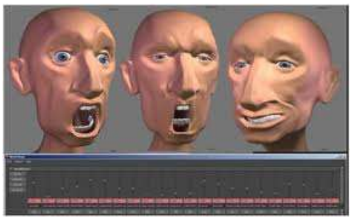
the “temporal resolution” of the animation

- keyframes distribution can be *adaptive*
  - more keyframes only where needed
- in-betweening happens on demand
  - e.g., at each refresh of video
- *times* associated to keyframe are arbitrary
  - not necessarily an integer number of video frames
  - all frames shown on screen will be in-betweens
- the better the interpolation schema
  → better in-betweens → fewer keyframes are needed
- editing the animation: ← asset
  - editing individual keyframes
  - editing keyframe *times* (e.g., to achieve non-linearity of moment, vary speed)
  - 1. pick a new time t_i (not a keyframe)
    2. bake the in-between at t as a new keyframe
    3. edit it!

34

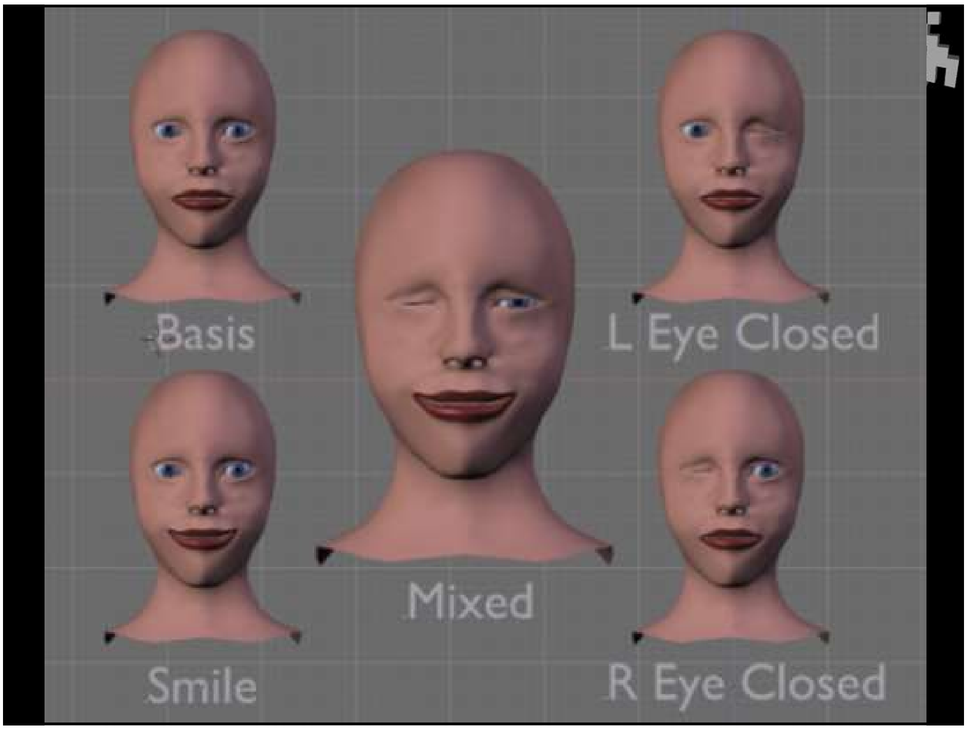
Uses of Blend shapes: facial animations



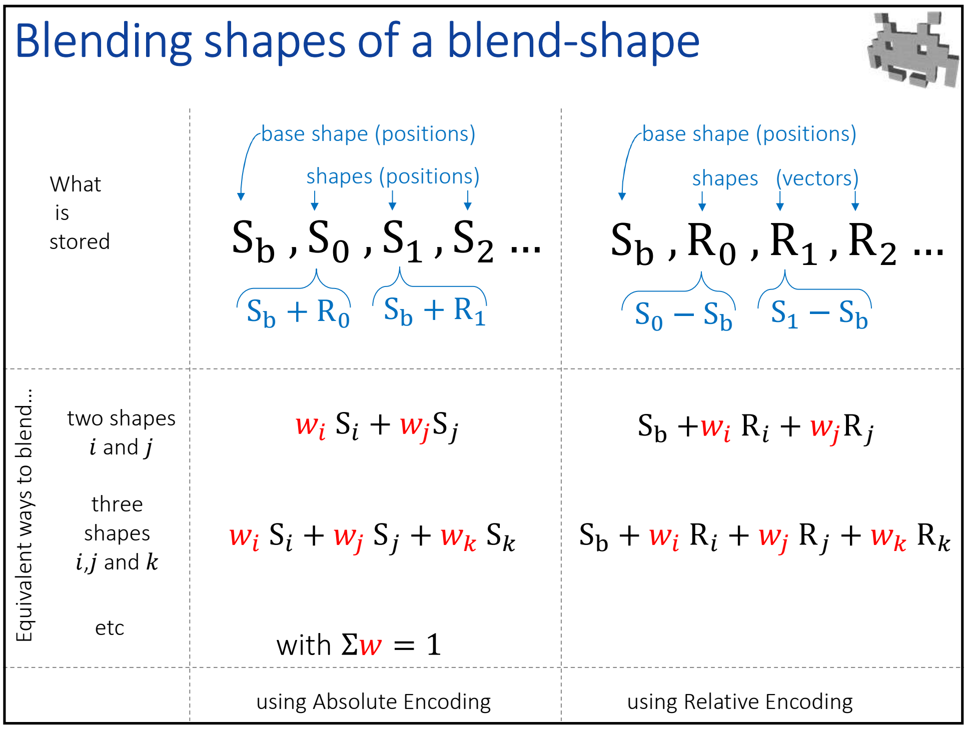


Here, used together with skeletal animations (see next lecture)
(for mandible, neck, eyeballs)

37



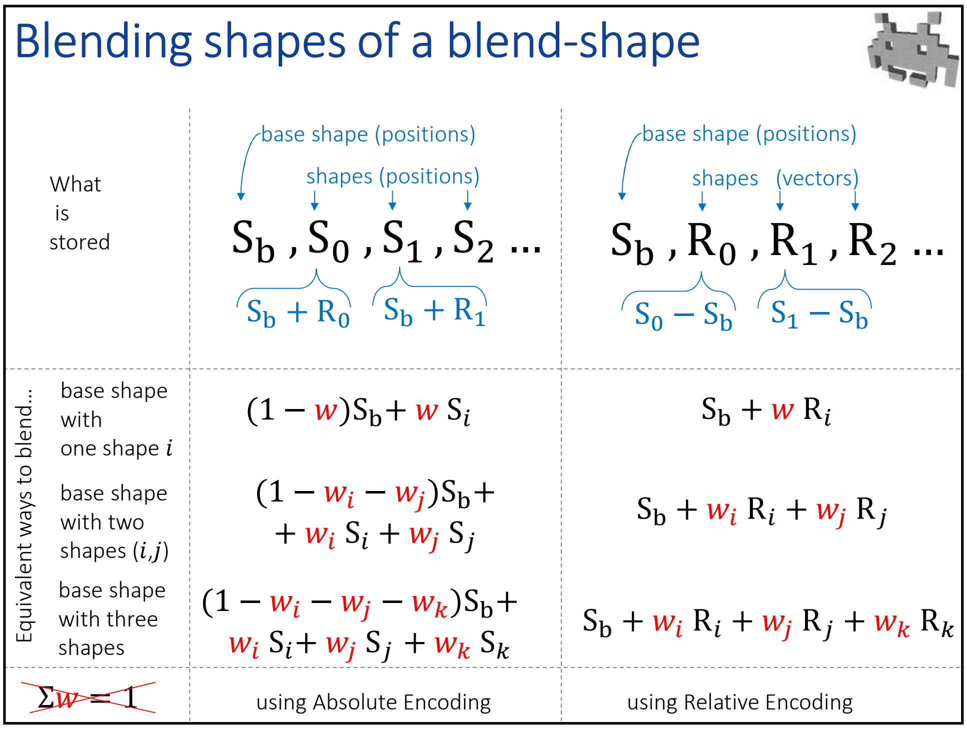
38



Blending shapes of a blend-shape

What is stored	base shape (positions) shapes (positions) $S_b, \underbrace{S_0}_{S_b + R_0}, \underbrace{S_1}_{S_b + R_1}, S_2 \ldots$	base shape (positions) shapes (vectors) $S_b, \underbrace{R_0}_{S_0 - S_b}, \underbrace{R_1}_{S_1 - S_b}, R_2 \ldots$
Equivalent ways to blend… two shapes i and j	$w_i S_i + w_j S_j$	$S_b + w_i R_i + w_j R_j$
three shapes i,j and k	$w_i S_i + w_j S_j + w_k S_k$	$S_b + w_i R_i + w_j R_j + w_k R_k$
etc	with $\Sigma w = 1$	
	using Absolute Encoding	using Relative Encoding

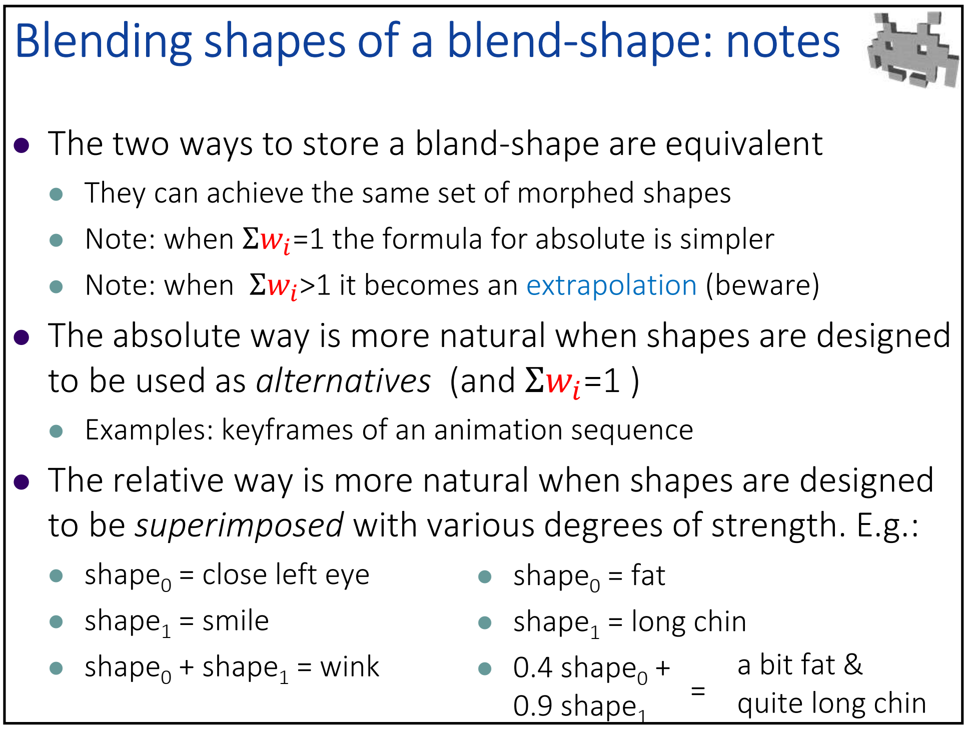
39



Blending shapes of a blend-shape

What is stored	base shape (positions) shapes (positions) $S_b , \underbrace{S_0}_{S_b + R_0} , \underbrace{S_1}_{S_b + R_1} , S_2 \dots$	base shape (positions) shapes (vectors) $S_b , \underbrace{R_0}_{S_0 - S_b} , \underbrace{R_1}_{S_1 - S_b} , R_2 \dots$
Equivalent ways to blend… base shape with one shape i	$(1 - w)S_b + w\, S_i$	$S_b + w\, R_i$
base shape with two shapes (i,j)	$(1 - w_i - w_j)S_b +$ $+ w_i\, S_i + w_j\, S_j$	$S_b + w_i\, R_i + w_j\, R_j$
base shape with three shapes	$(1 - w_i - w_j - w_k)S_b +$ $w_i\, S_i + w_j\, S_j + w_k\, S_k$	$S_b + w_i\, R_i + w_j\, R_j + w_k\, R_k$
$\cancel{\Sigma w = 1}$	using Absolute Encoding	using Relative Encoding

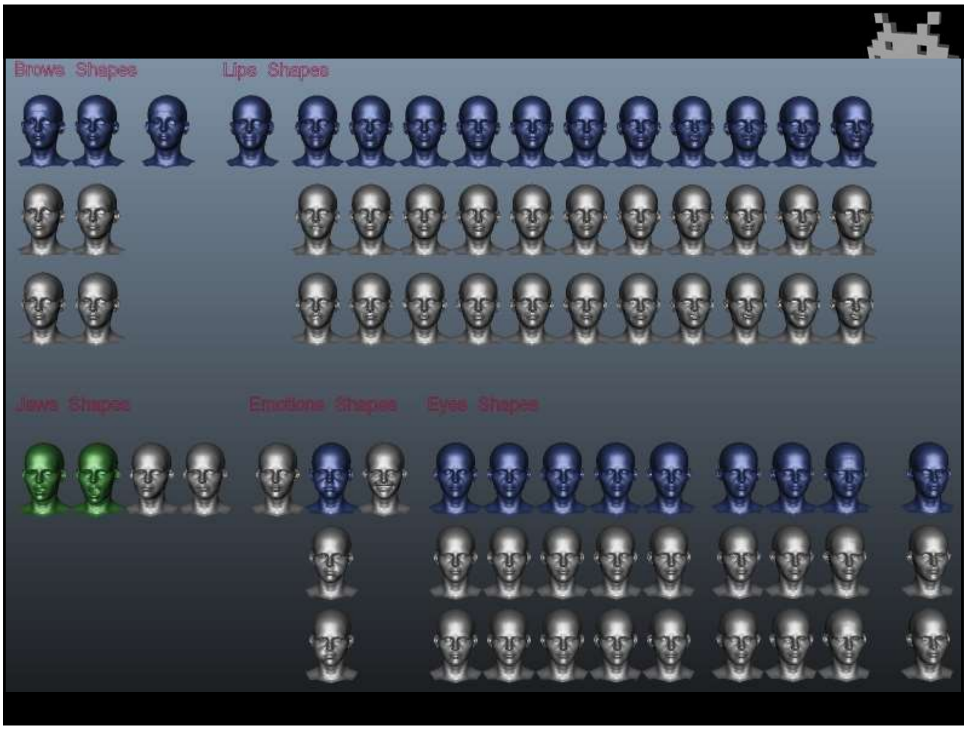
40



Blending shapes of a blend-shape: notes

- The two ways to store a bland-shape are equivalent
  - They can achieve the same set of morphed shapes
  - Note: when Σw_i=1 the formula for absolute is simpler
  - Note: when Σw_i>1 it becomes an extrapolation (beware)
- The absolute way is more natural when shapes are designed to be used as *alternatives* (and Σw_i=1)
  - Examples: keyframes of an animation sequence
- The relative way is more natural when shapes are designed to be *superimposed* with various degrees of strength. E.g.:
  - shape$_0$ = close left eye
  - shape$_1$ = smile
  - shape$_0$ + shape$_1$ = wink
  - shape$_0$ = fat
  - shape$_1$ = long chin
  - 0.4 shape$_0$ + 0.9 shape$_1$ = a bit fat & quite long chin

41

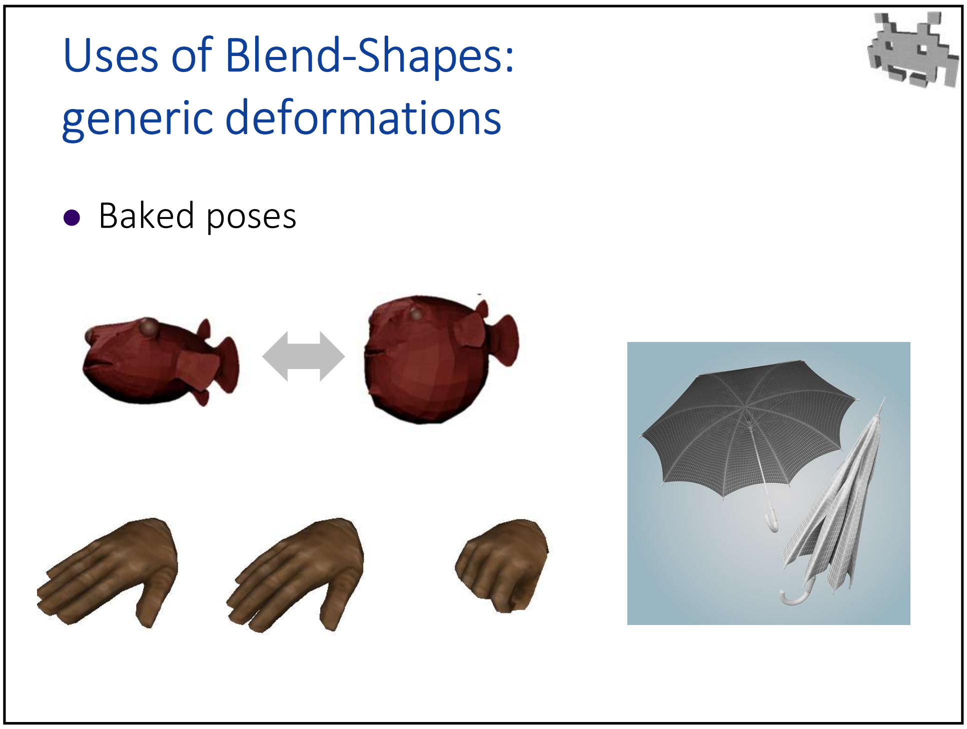


45

Using facial animations as Blend shapes

- 3D Modeller authors:
  produces the blend-shapes (aka: the “facial rig”)
- Animator (of expressions) picks:
  weights
  - eg.: with sliders
  - assisted / substituted by automatisms
    - e.g., lip sync
    - e.g., dynamically determined expressions
- Key-shape Blending: task of the rendering engine

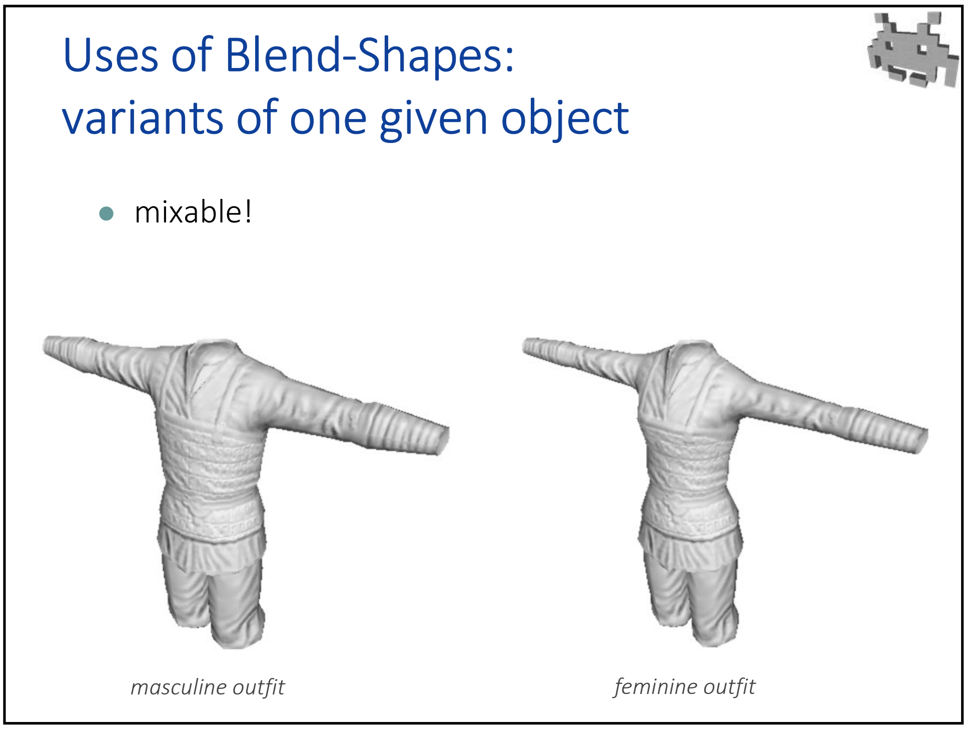
46



Uses of Blend-Shapes: generic deformations

- Baked poses

47



Uses of Blend-Shapes: variants of one given object

- mixable!

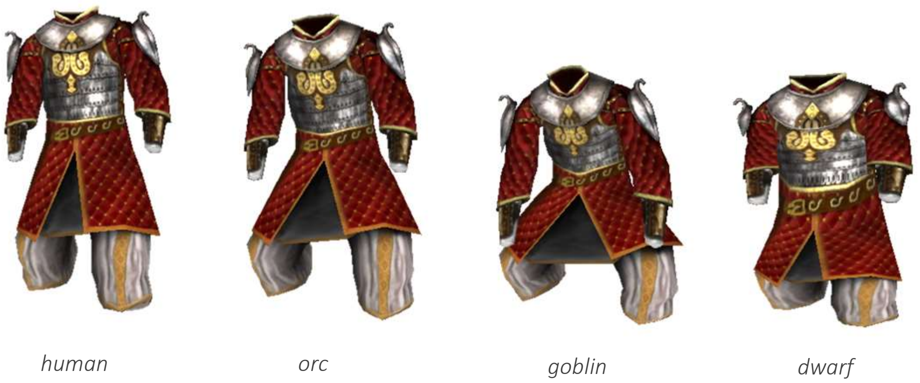
masculine outfit

feminine outfit

48

Uses of Blend-Shapes
variants of one given object

- mixable!



49

Uses of Blend-Shapes

- Defines shapes of a class of objects
  - get a shape in the class = just choose the weights
    - 3D modelling at a high-level of abstraction
  - the weights “span” one shape space
    - one given shape = one point in the space
    - weights = coords
  - the space is the more useful the more:
    - *all and only* the reasonable shapes are represented in the space
- Typical Example: face morphologies
  - “face-space”
  - note: face morphology ≠ facial expression

50

Uses of Blend shapes

- A blend shape modelling a face space (“face-morphs”)

51

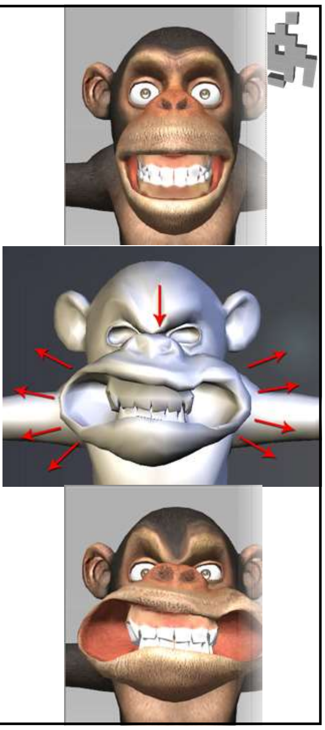
All morph-shape share…
(so, a blend-shape *cannot* change)…

- The mesh connectivity
  - Eg. no change mesh res, remeshing
- Therfore, the surface topology
  - E.g. no breaking apart, fusing parts
- The mesh attributes
  - Such as color, UV-map…
  - Exceptions: positions, normals
- The textures
  - Use a texture animation instead?

52

Blend shapes: authoring

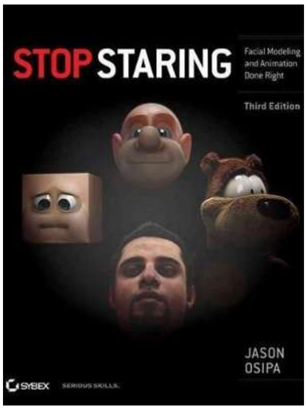
1. Editing base shape
   - including: uv-mapping, texturing, etc.
2. Re-edit it for each shape-key! …while preserving: connectivity, textures, etc:
   - with low poly editing
   - or with subdivision surfaces…
   - or with parametric surfaces…
   - or with scupting.



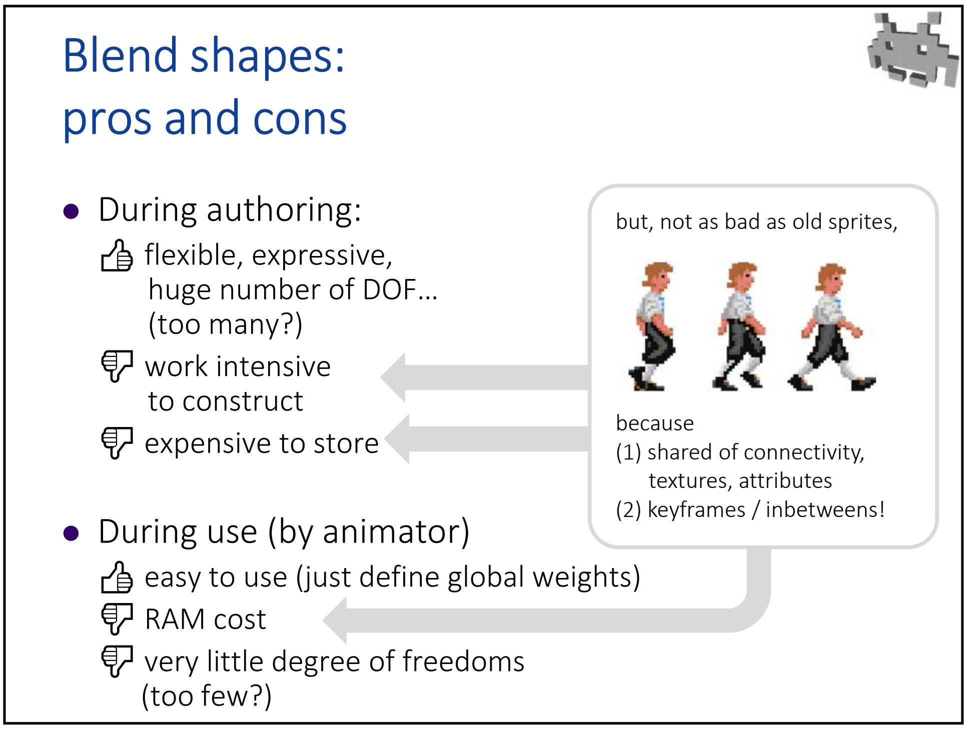
53

Blend shapes: authoring

- Handbook for blend-shape based face animation:
  - “Stop Staring” (3d edition) Jason Osipa
  - Covers: style, expression…
  - Non technical (high level)
  - Not about specific tools e.g. Blender, Maya



54



Blend shapes: pros and cons

- During authoring:
  - 👍 flexible, expressive, huge number of DOF… (too many?)
  - 👎 work intensive to construct
  - 👎 expensive to store
- During use (by animator)
  - 👍 easy to use (just define global weights)
  - 👎 RAM cost
  - 👎 very little degree of freedoms (too few?)

but, not as bad as old sprites,

because
(1) shared of connectivity, textures, attributes
(2) keyframes / inbetweens!

56

Blend shapes: open challenges

- Capturing:
  - from a stream of meshes
  - e.g. : from a RGBD camera (like Microsoft Kinect) to a blend-shape: difficult!
- Compression
  - e.g.: reduce number of keyframes (can you think of an algorithm?)
- Streaming
  - server sends animation to client while it runs
- LOD-ding
  - like for meshes (but more difficult)

57