A funny glitch: https://www.youtube.com/watch?v=Nob6kcLod3w&feature=youtu.be

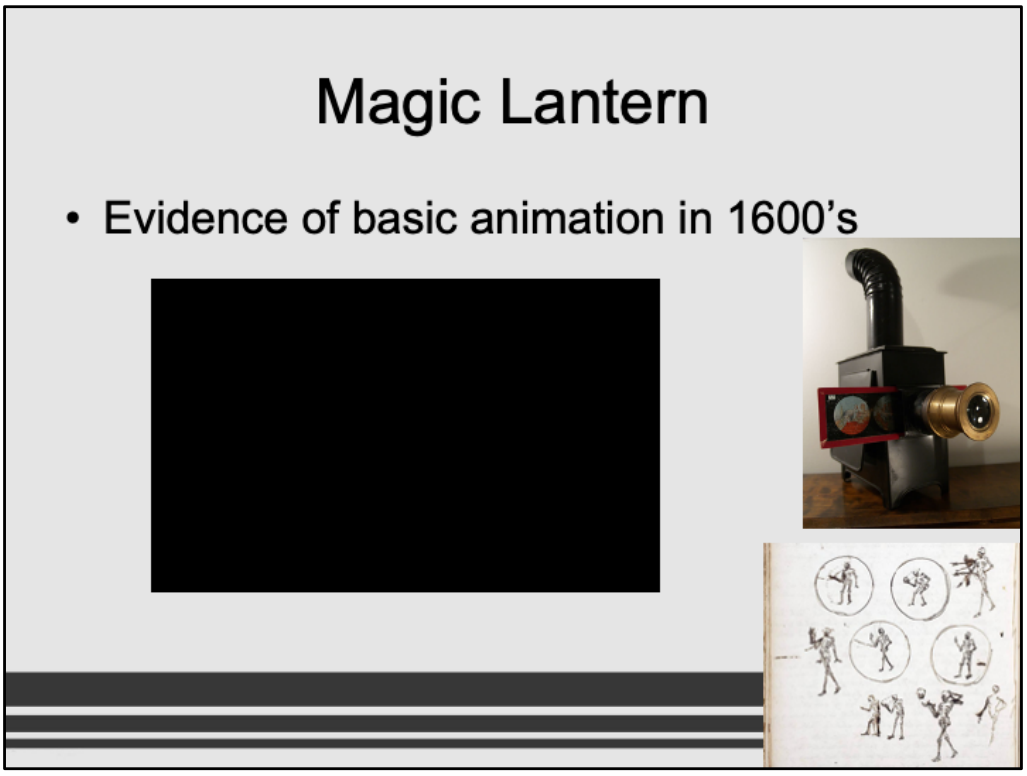
EARLY NON-INTERACTIVE ANIMATION

Cave Art and Shadow Play

- Cave art with signs of motion
  - Speculation that flickering fire randomly illuminated different parts of image (e.g. legs appear to move)
- Shadow play developed in many cultures
- Shadow cast by light of puppets

Puppetry

- Perhaps 4000 years old



Plans for animation of skeleton in 1659

https://www.youtube.com/watch?v=t0waEVSXpYA

Note that video depicts animated magic lantern animations much more recent than 1600's

Automata

- Likely 1000's of years old techniques
- Machine animation
- Gears, levers, clockwork, etc.
- Spring power, or other mechanical source
- Disney later developed Animatronics



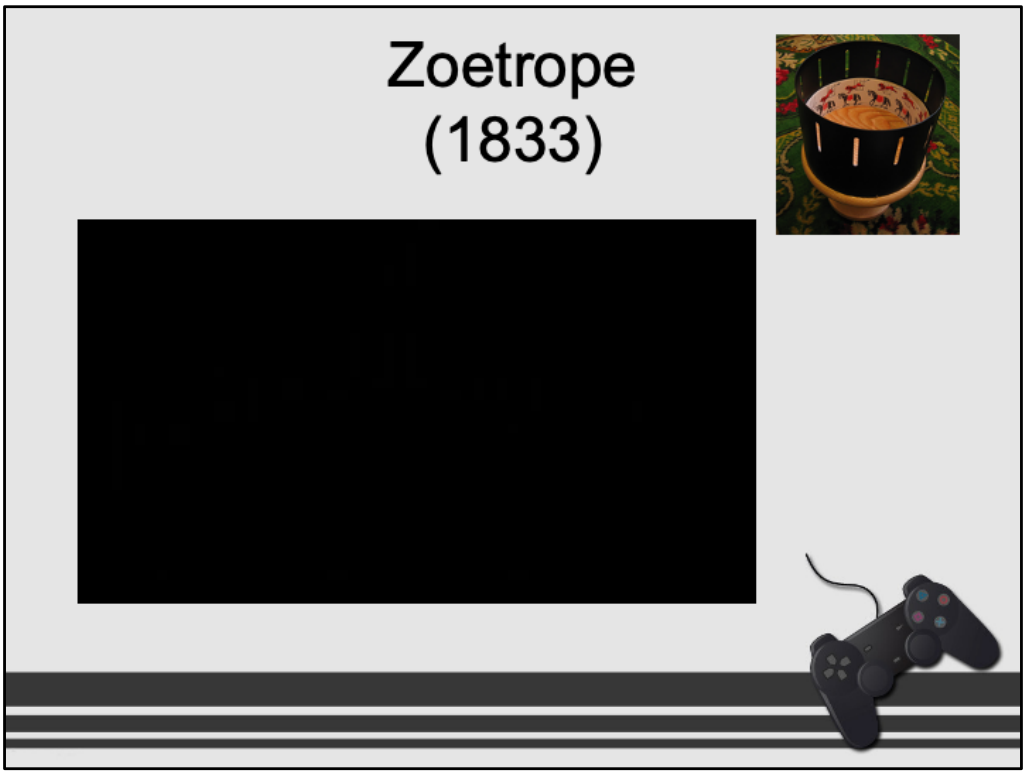


https://www.youtube.com/watch?v=kd826ZXeGOo

Look through slits in a mirror

https://www.youtube.com/watch?v=3JeN3uk2ClE

https://www.youtube.com/watch?v=UqwkdlwmHig



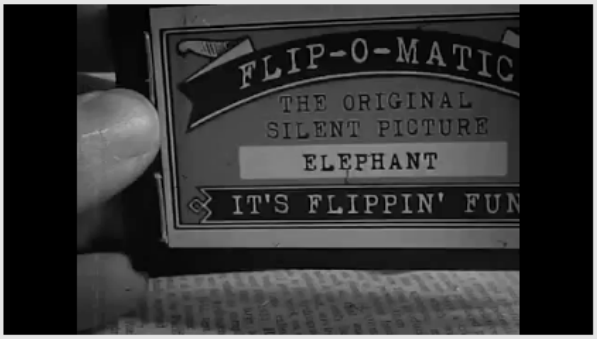
https://www.youtube.com/watch?v=SBg6dAE3mI0

https://www.youtube.com/watch?v=-66v1ARI0-Q

Kineograph (1868)

- Flipbook

Kineograph, 1868



https://www.youtube.com/watch?v=7K3aMdnCqXg

Early Film

- Arrival of a Train at La Ciotat (1895)

https://www.youtube.com/watch?v=1dgLEDdFddk

See-oh-tah (”t” at end silent)

Early Cartoon

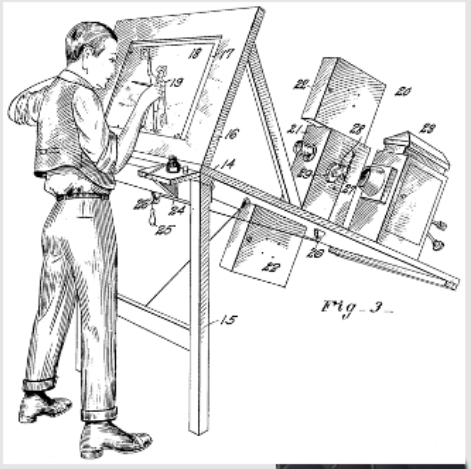
- Fantasmagorie (1908)

https://www.youtube.com/watch?v=swh448fLd1g

1908

Rotoscope

- Max Fleischer (1915 Patent)
- Video frame projected onto glass
- Sketched over



Out of the Inkwell

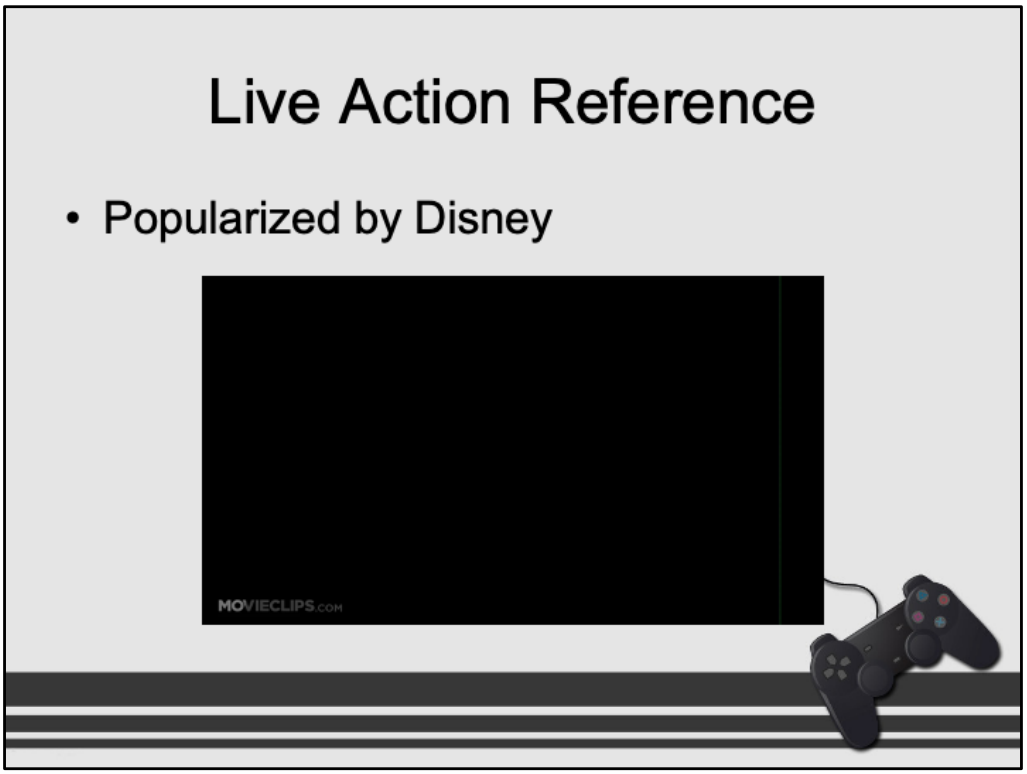
- First use of rotoscoping
- KoKo the Clown

Disney – Snow White

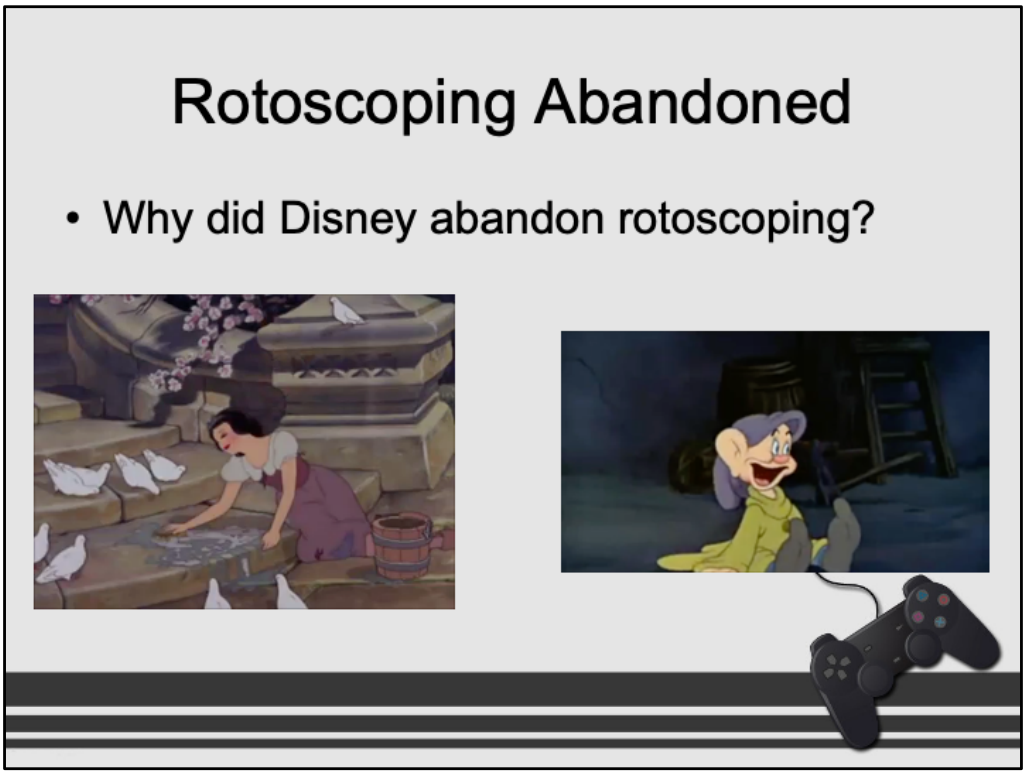
- Rotoscoped w/ Actress, Marge Champion

https://www.youtube.com/watch?v=4Wgm2OzDYVo

Marge Champion - https://www.youtube.com/watch?v=Hjt13WdEkZc



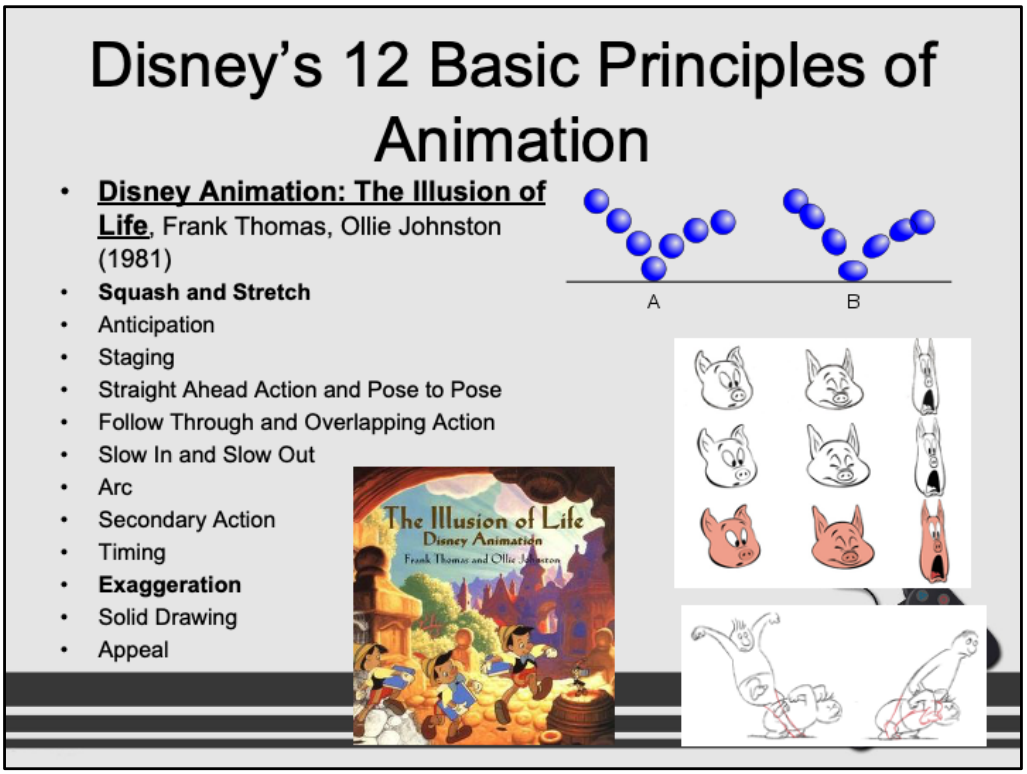
https://www.youtube.com/watch?v=LWwO-h7ZSlw



Snow white versus the 7 dwarfs

Snow white well song
https://www.youtube.com/watch?v=54QeNL5ih6A

The washing song
https://www.youtube.com/watch?v=ziyudbAiWrI



https://ohmy.disney.com/movies/2016/07/20/twelve-principles-animation-disney/

https://en.wikipedia.org/wiki/12_basic_principles_of_animation

Making Of Disney's Snowwhite or How Disney Cartoons are made:
https://www.youtube.com/watch?v=OebUzEhSLBI

https://ohmy.disney.com/movies/2016/07/20/twelve-principles-animation-disney/

Animators
Onion skin

Moviola – scrub through early animations

Inking on celluloid

Stage Settings

Foley Sound effects

Soundtrack

EARLY INTERACTIVE ANIMATION

Interactive Automata

- Penny Arcades
- Early Simulators

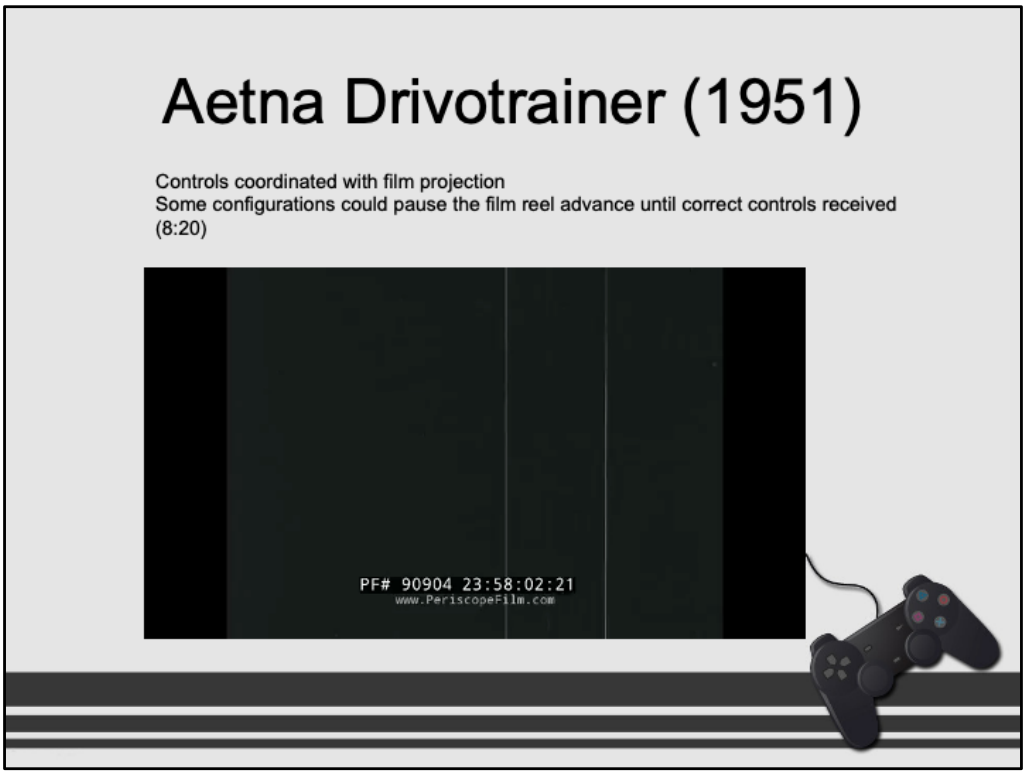
Interactive Automata Games

- Penny Arcade
  - Bowling, baseball, etc.
- Board Games
  - Hungry Hungry Hippos
  - Rock 'em Sock 'em Robots

https://www.youtube.com/watch?v=owwJ9Y-Wbzw

Simulators

- Link Trainer (WWII)
- Dehmel Flight Simulator
- Aerostructor
- Aetna Drivotrainer
- Etc.



https://www.youtube.com/watch?v=jTcCNQ7T4rY

Link trainer (flight) at beginning

1:00 look at the Drivotrainer classroom setup

2:48 scoring machine

8:20 interactive video playback

INTERACTIVE ANIMATION IN VIDEO GAMES

Vector Graphics

- Tennis for Two (1958)
- Spacewar! (1962)
- Geometric transformations of wireframe

Tennis for Two – first analog video game

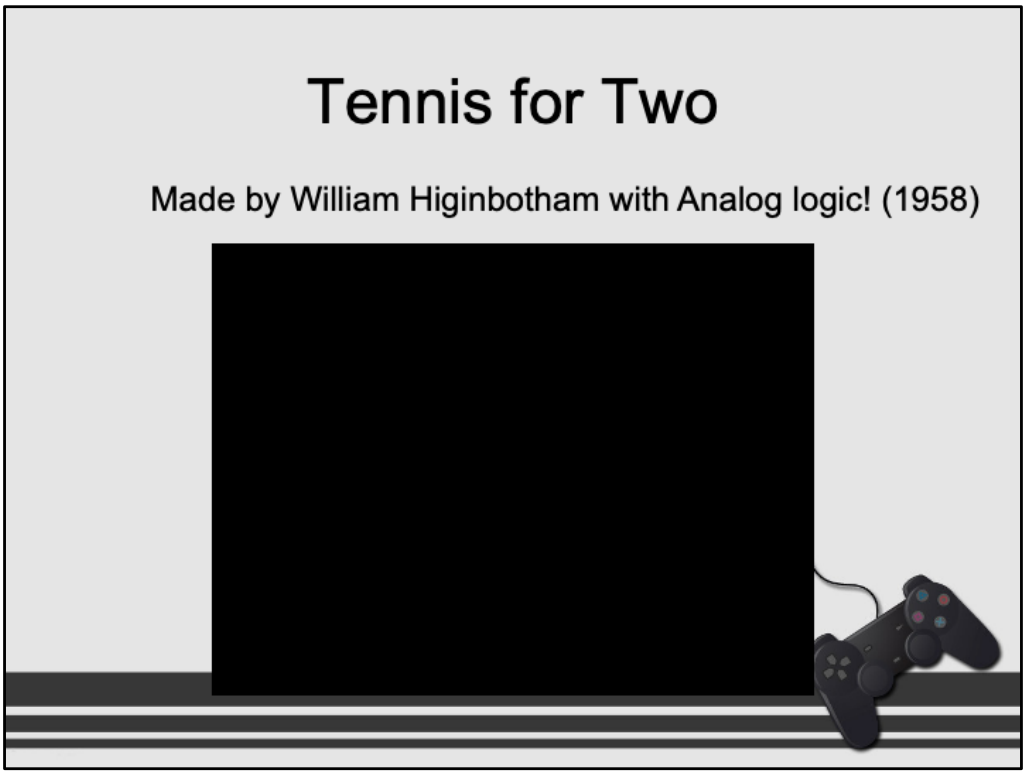
Spacewar! – first digital video game

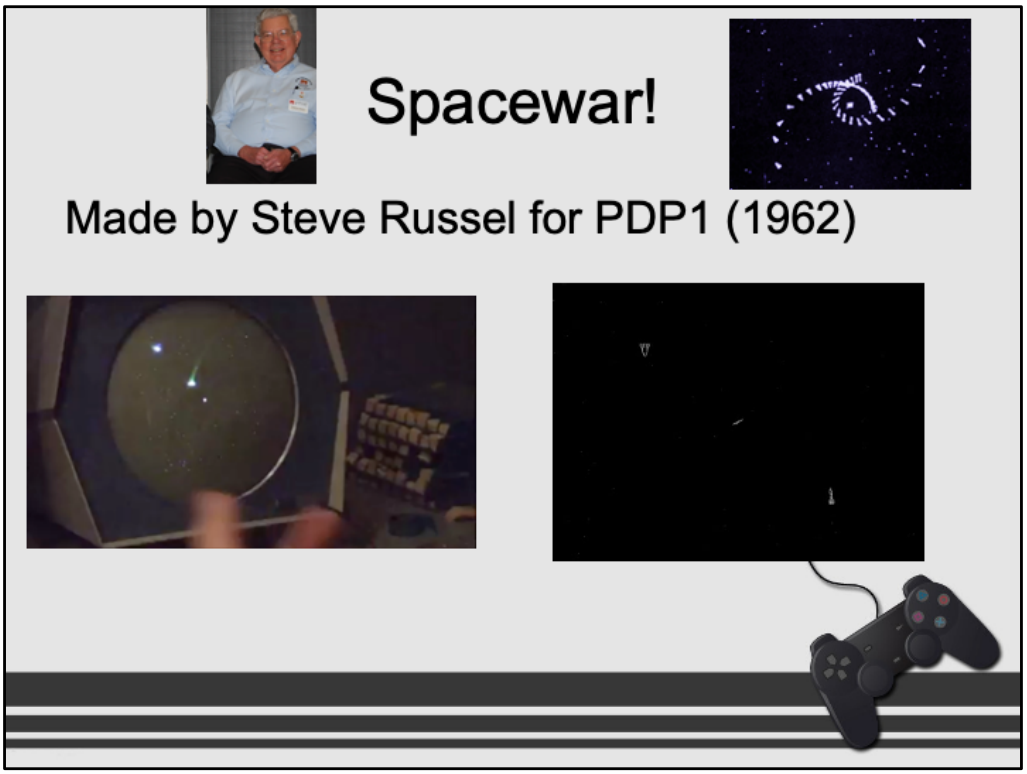
First video game

https://www.youtube.com/watch?v=Rmvb4Hktv7U

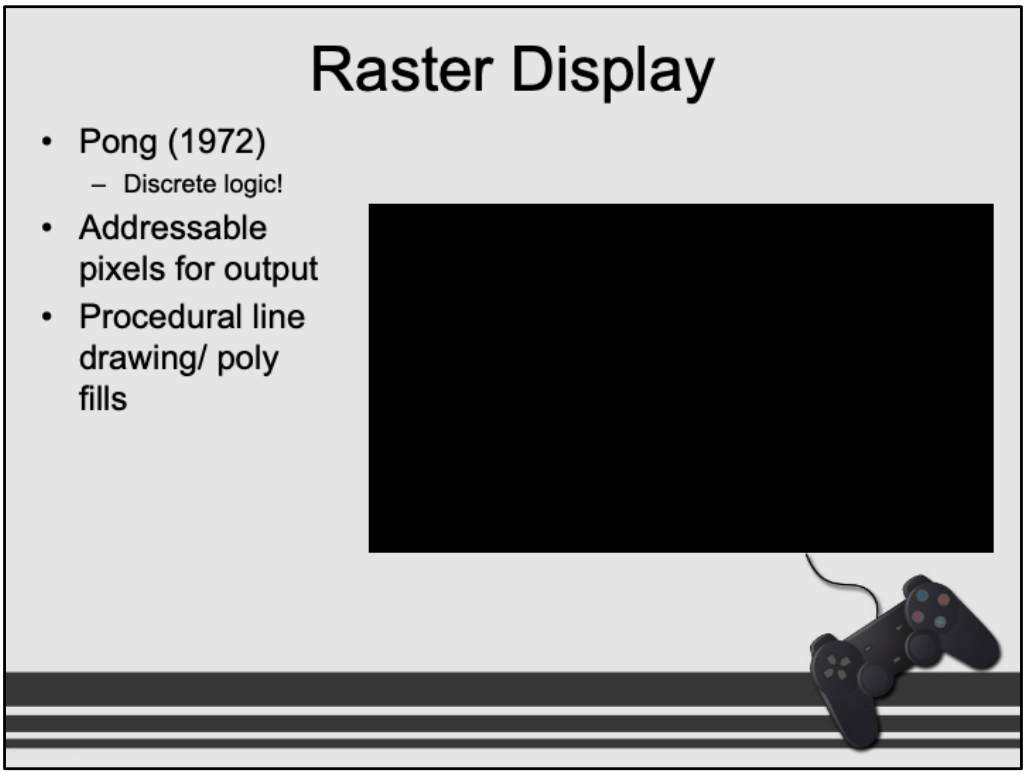
Tennis for Two

Made by William Higinbotham with Analog logic! (1958)





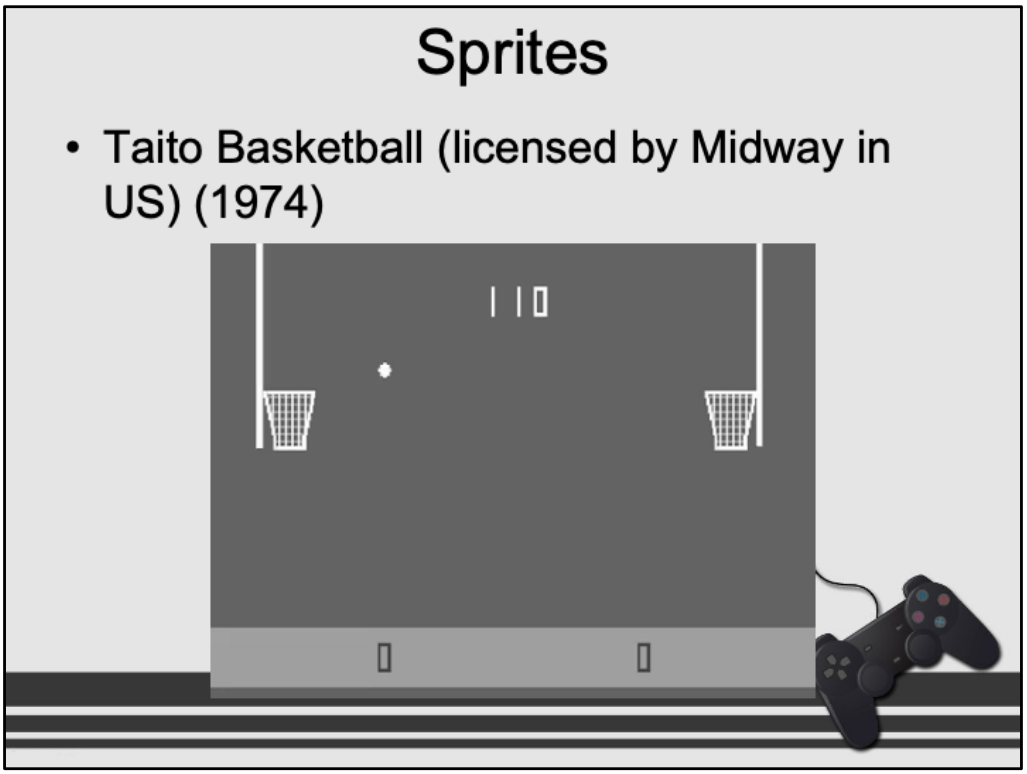
https://www.youtube.com/watch?v=7bzWnaH-0sg



https://www.youtube.com/watch?v=fiShX2pTz9A

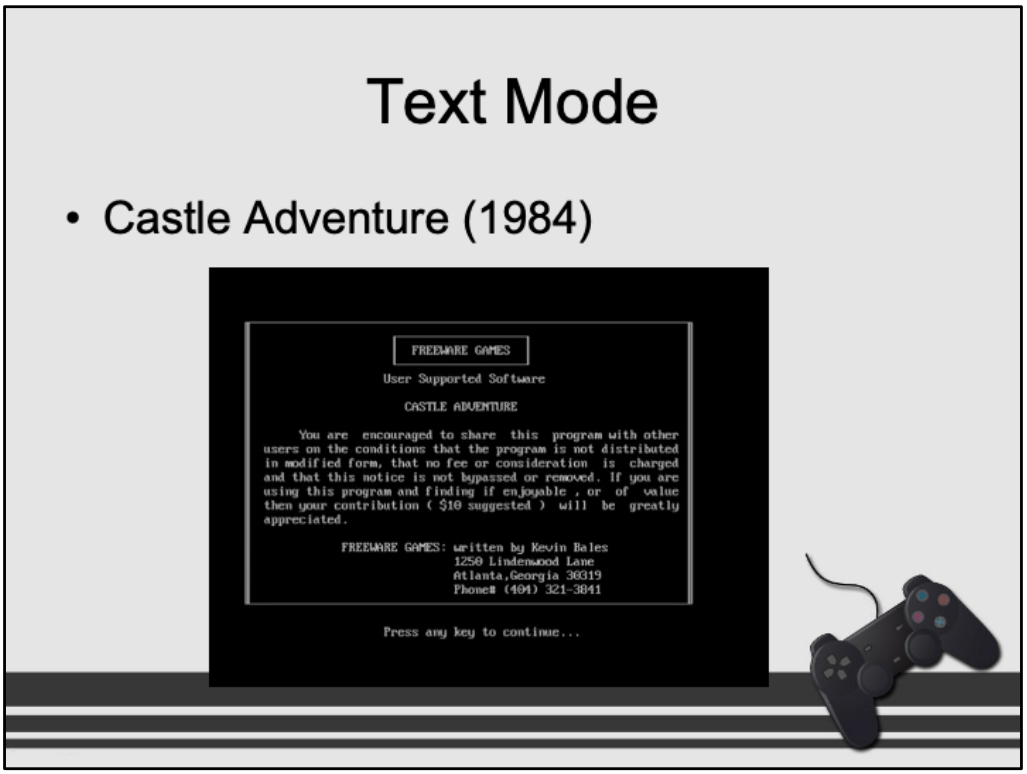
Discrete logic – not a central processing unit. Individual logic gates physically distributed

Discrete logic used for a number of arcade machines as much cheaper at the time compared to using a CPU



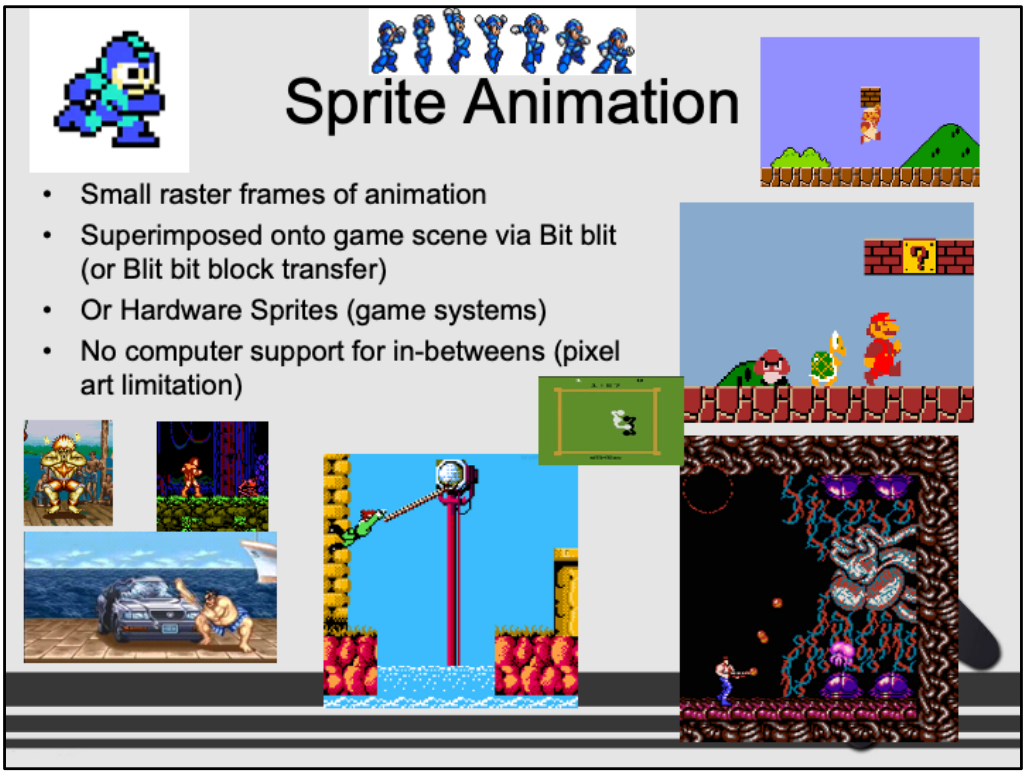
Arrays of pixel data that are copied to output pixel locations

https://www.youtube.com/watch?v=BMV_meVa7m8



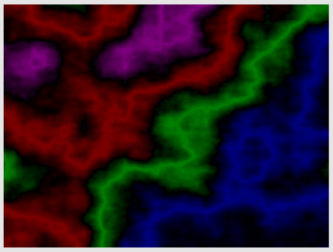
Castle adventure: (1984)

https://www.youtube.com/watch?v=OkPr_Xz6Ftk


Sprite Animation
• Small raster frames of animation
• Superimposed onto game scene via Bit blit (or Blit bit block transfer)
• Or Hardware Sprites (game systems)
• No computer support for in-betweens (pixel art limitation)

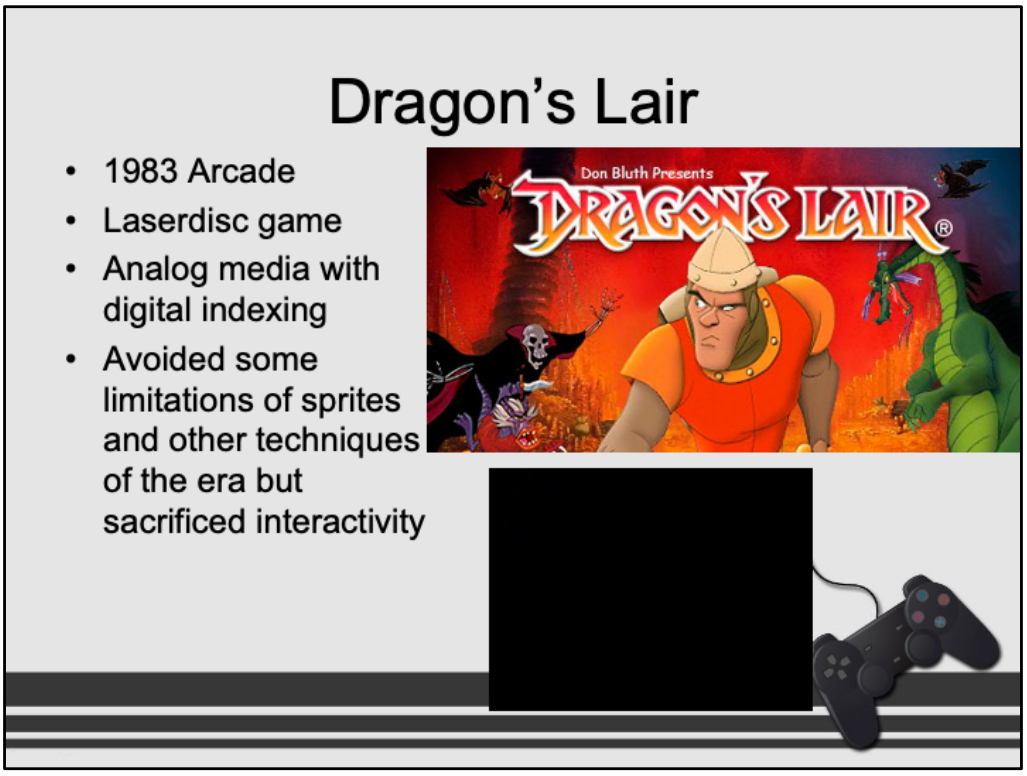
Color Cycling

- Animated Color Palette
- Graphics modes that use palette (color lookup table)



http://www.effectgames.com/effect/article-Old_School_Color_Cycling_with_HTML5.html

http://www.effectgames.com/demos/canvascycle/



A game that is really closely tied to fundamentals of hand drawn animation, but introduces interactivity

What are the drawbacks of the approach taken by Dragon's Lair?

Only "Quick Time Events"

Laserdisc dump of dragon's lair

https://www.youtube.com/watch?v=znO_m00s8II

Don Bluth

- 80's animator
- Game Designer
- The Secret of NIMH
- An American Tale
- The Land Before Time
- Dragon's Lair
- Space Ace
- Etc.

Don Bluth

- 80's animator
- Game Designer
- The Secret of NIMH
- An American Tale
- The Land Before Time
- Dragon's Lair
- Space Ace
- Etc.

Donald Knuth – Famous CS Professor from Standford

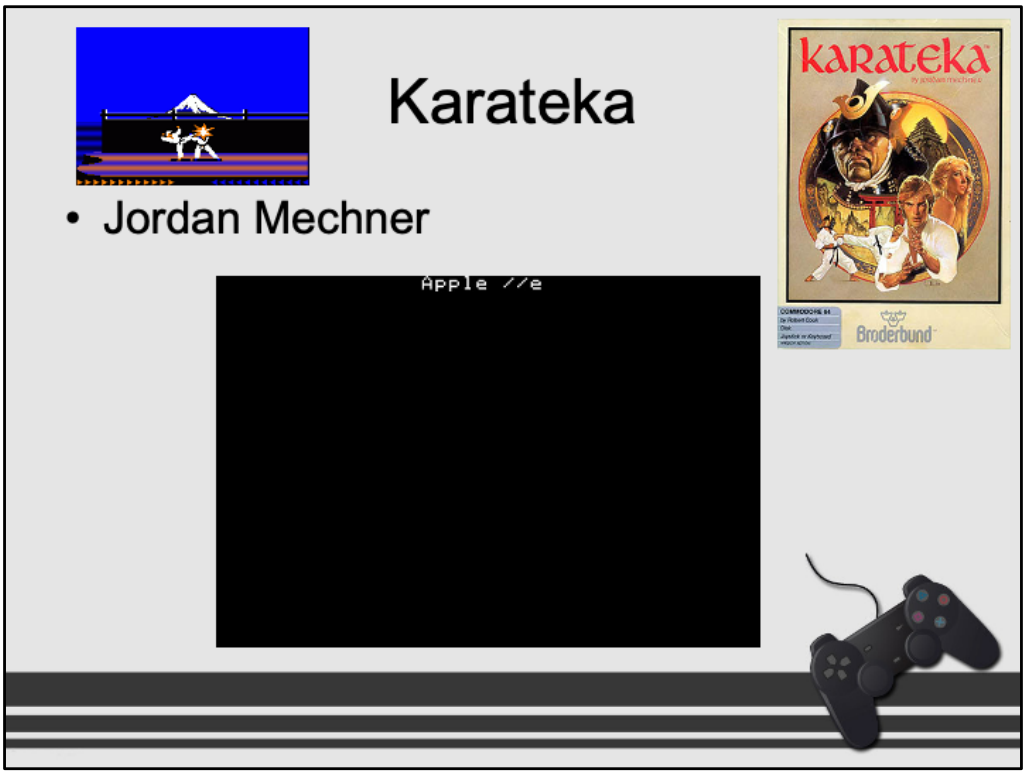
Don Bluth

- 80's animator
- Game Designer
- The Secret of NIMH
- An American Tale
- The Land Before Time
- Dragon's Lair
- Space Ace
- Etc.
- (Completely different guy than **Donald Knuth**)

Donald Knuth – Famous CS Professor from Standford

Rotoscoped Gaming

- Karateka (1984)
- Prince of Persia (1989)
- Dragon's Lair (1989 Amiga port of arcade)
- Another World (aka Out of this World) (1991)
- Flashback (1992)
- The Last Express (1997)



Mechner says: CARE – ah – TEK – ah

But thinks any pronounciation is fine

https://www.youtube.com/watch?v=wKqk9kosCs4

1984

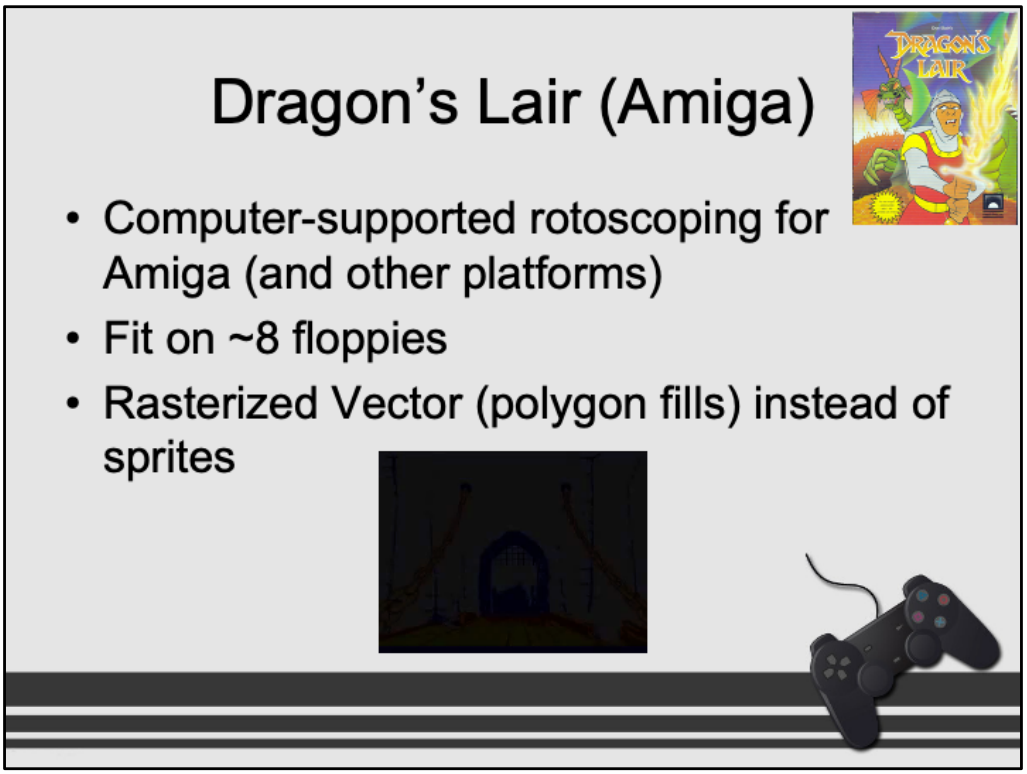
1989 Apple II version

2:44 level starts

Erol Flynn – The Adventures of Robin Hood

http://www.museumofplay.org/blog/chegheads/2014/10/jordan-mechner-collection-documents-revolution-in-game-graphics/

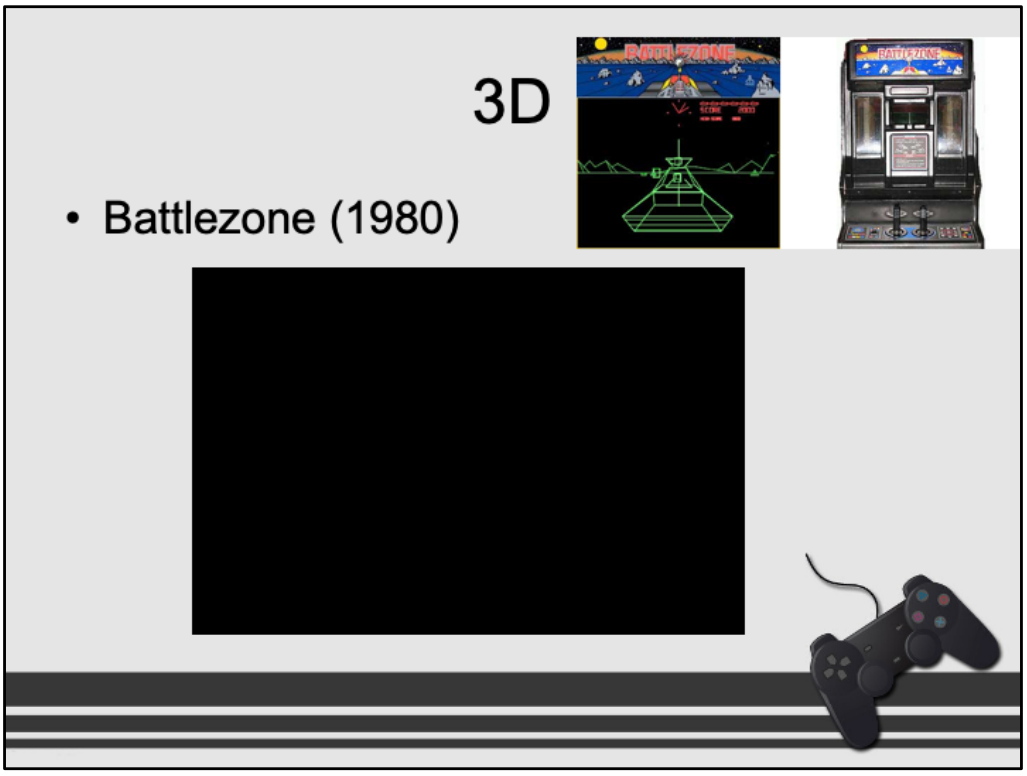
https://www.youtube.com/watch?v=fggouSd3dr4



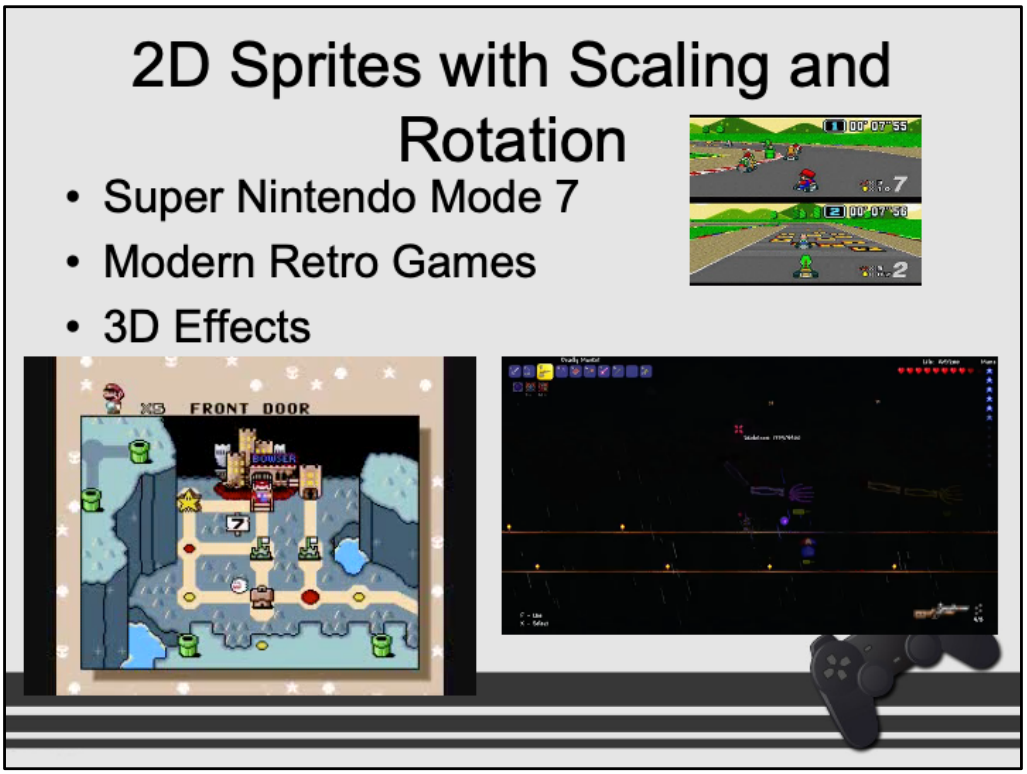
Amiga Version

https://www.youtube.com/watch?v=dSE5LFdOtQI

https://www.youtube.com/watch?v=GTb6zX7Hd5I



Rendered on a Vector CRT Display (not raster)



Super Mario world

https://www.youtube.com/watch?v=UToiJXdycog

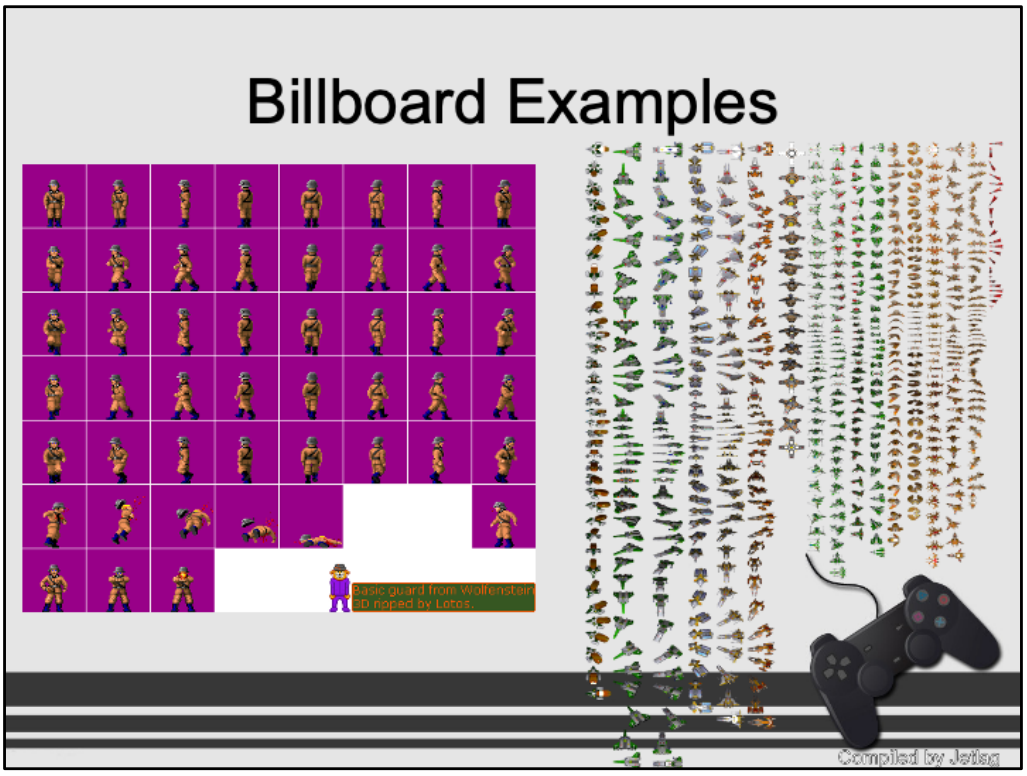
Mode 7 is a special background texture rendering pipeline

Terraria Example vid: (warning commentary that may be inappropriate)

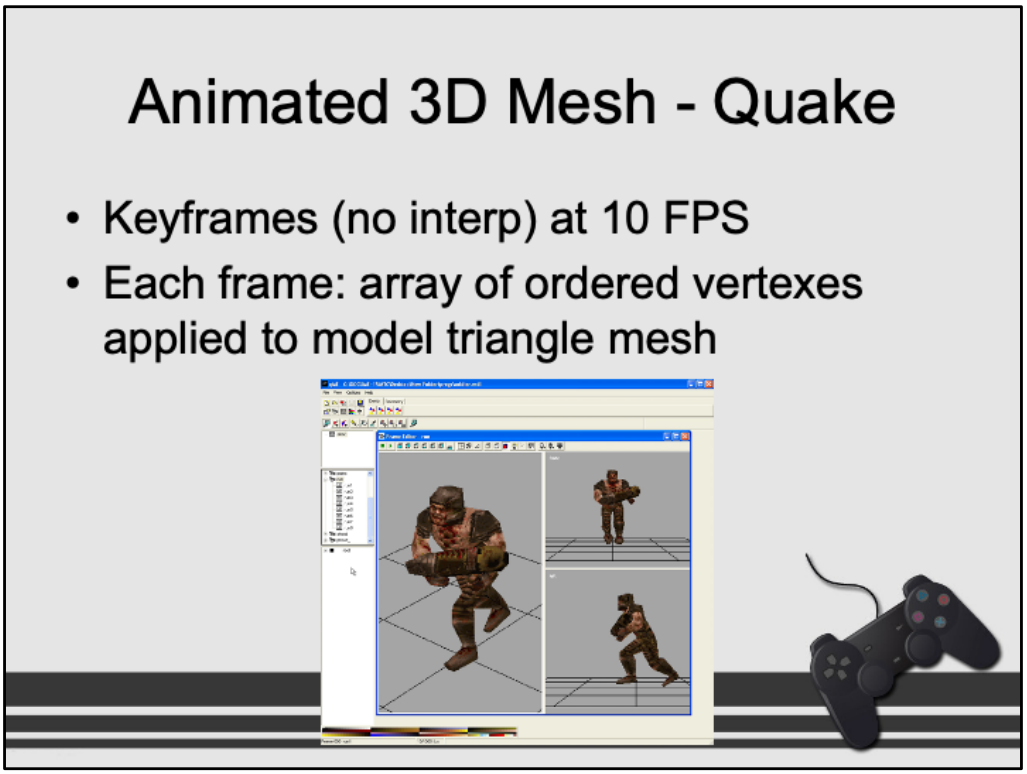
https://www.youtube.com/watch?v=jsxILQvZcr4

3D with 2D Billboards

- Wolfenstein, Doom
- Wing Commander



Wolfenstein3d and Wing Commander



4D Sports Boxing – 1991 (mo-cap, unclear if interpolated)

Alone in the Dark – 1992 (had interpolation)

I captured from Quake Model Explorer (qME) run on a VM. Not hosted online, sorry

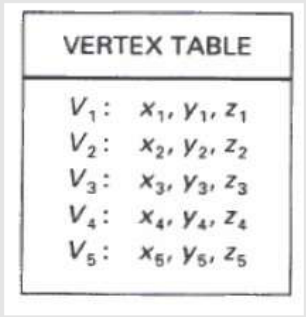
Animated 3D Mesh Rendering

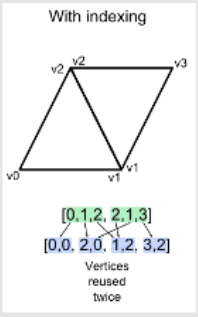
- Identify curr frame
- Use model's triangles to obtain vertex indexes of curr frame's vertices

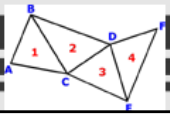
```
struct mdl_vertex_t *pvert1, *pvert2;
vec3_t v;

for (/* ... */)
  {
    pvert1 = &mdl->frames[current].frame.verts[mdl->triangles[i].vertex[j]];
```

Animated 3D Mesh Storage

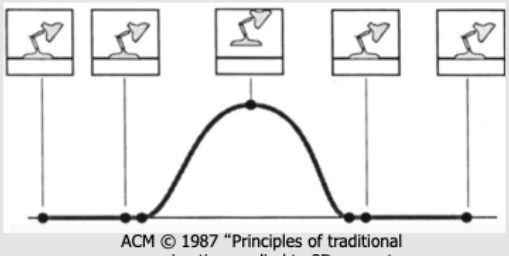
- Model consists of vertex (*x, y, z*) indexed triangles (*v1, v2, v3*) (or perhaps triangle strips w/ shared vertex indexes) and texture coordinate refs (*u,v*)
- Animation frames contain vertex tables with differing position vectors
- Animation played by parsing model tris with ref to curr frame verts
- Animation storage *increases* as model mesh detail *increases*!
- Increasing frame rate introduces an entire new vertex table per additional frame!







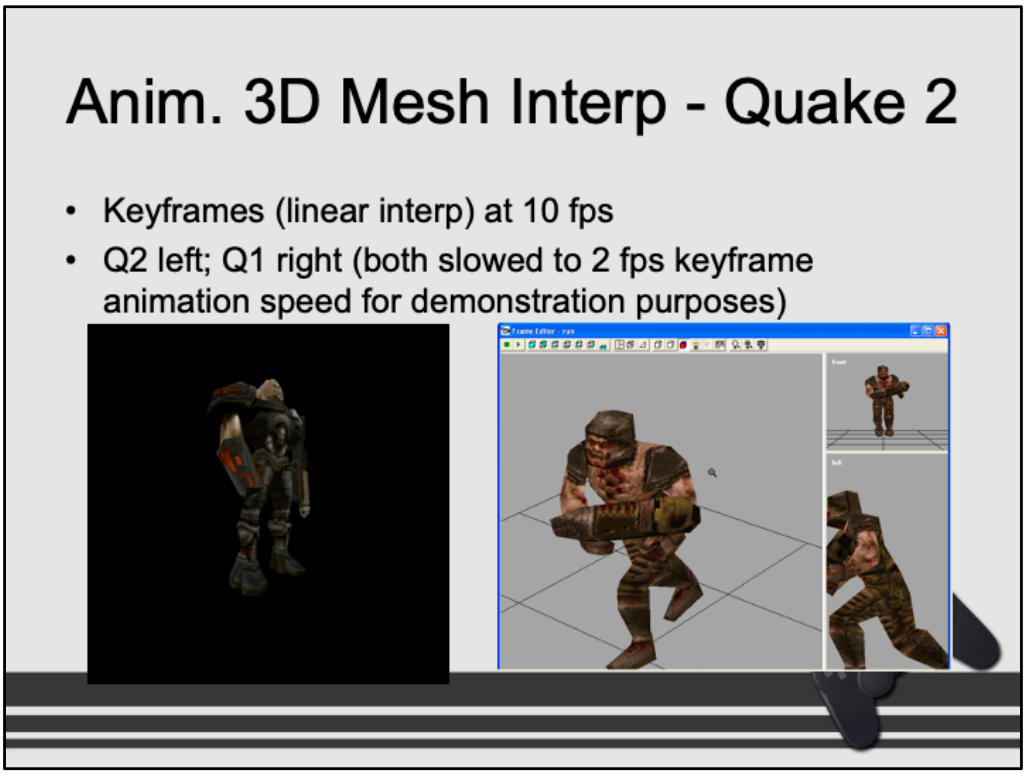
Computer-Assisted Animation



ACM © 1987 "Principles of traditional animation applied to 3D computer animation"

Keyframing

- automate the inbetweening
- good control
- less tedious
- creating a good animation still requires considerable skill and talent



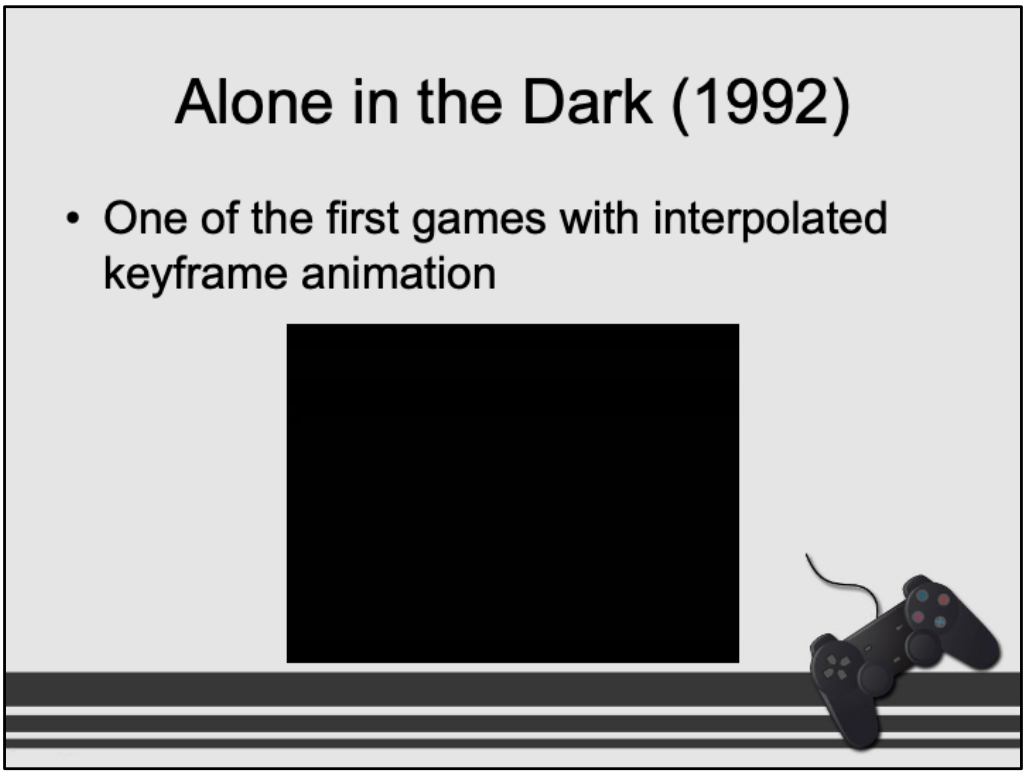
Quake 1 did not have inbetween frames (linear interp'ed) originally. Quake 2 however, did.

Not actually in-game animation, but appropriate for demonstation purposes
https://www.youtube.com/watch?v=7WYSS5Zz2xs

Interpolation

- Like before but obtain curr frame and next frame
- Linear Interpolate (lerp) between the two

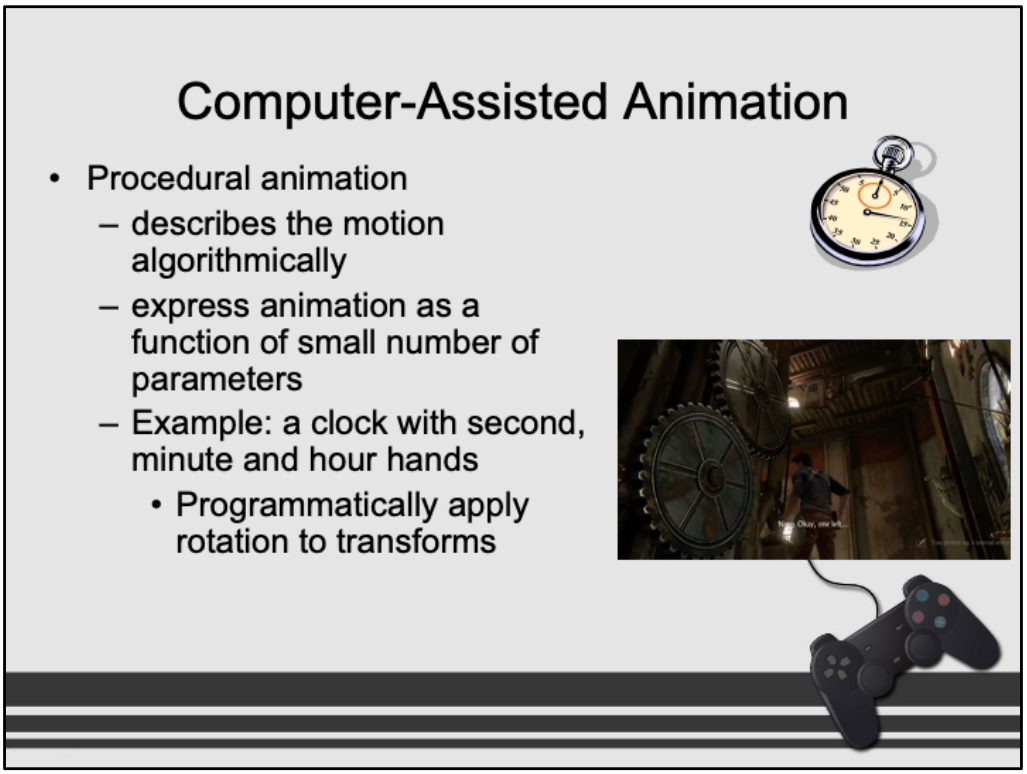
```
struct mdl_vertex_t *pvert1, *pvert2;
vec3_t v;

for (/* ... */)
  {
    pvert1 = &mdl->frames[current].frame.verts[mdl->triangles[i].vertex[j]];
    pvert2 = &mdl->frames[current + 1].frame.verts[mdl->triangles[i].vertex[j]];

    /* ... */

    v[0] = mdl->header.scale[0] * (pvert1->v[0] + interp * (pvert2->v[0] - pvert1->v[0])) + mdl->header.translate[0];
    v[1] = mdl->header.scale[1] * (pvert1->v[1] + interp * (pvert2->v[1] - pvert1->v[1])) + mdl->header.translate[1];
    v[2] = mdl->header.scale[2] * (pvert1->v[2] + interp * (pvert2->v[2] - pvert1->v[2])) + mdl->header.translate[2];

    /* ... */
  }
```



H. P. Lovecraft inspired game. Cthulhu

Gameplay a bit like resident evil

Different video that embedded but close enough:
https://www.youtube.com/watch?v=iSwYY2eoKhQ



Uncharted 4 – Clock Tower

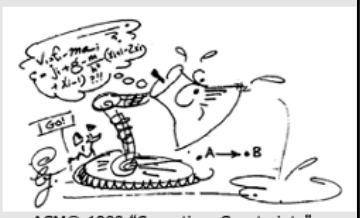
Probably different video, but close enough:

https://www.youtube.com/watch?v=oIWkkn_y_t8

Computer-Assisted Animation

- Physically Based Animation
  - Assign physical properties to objects

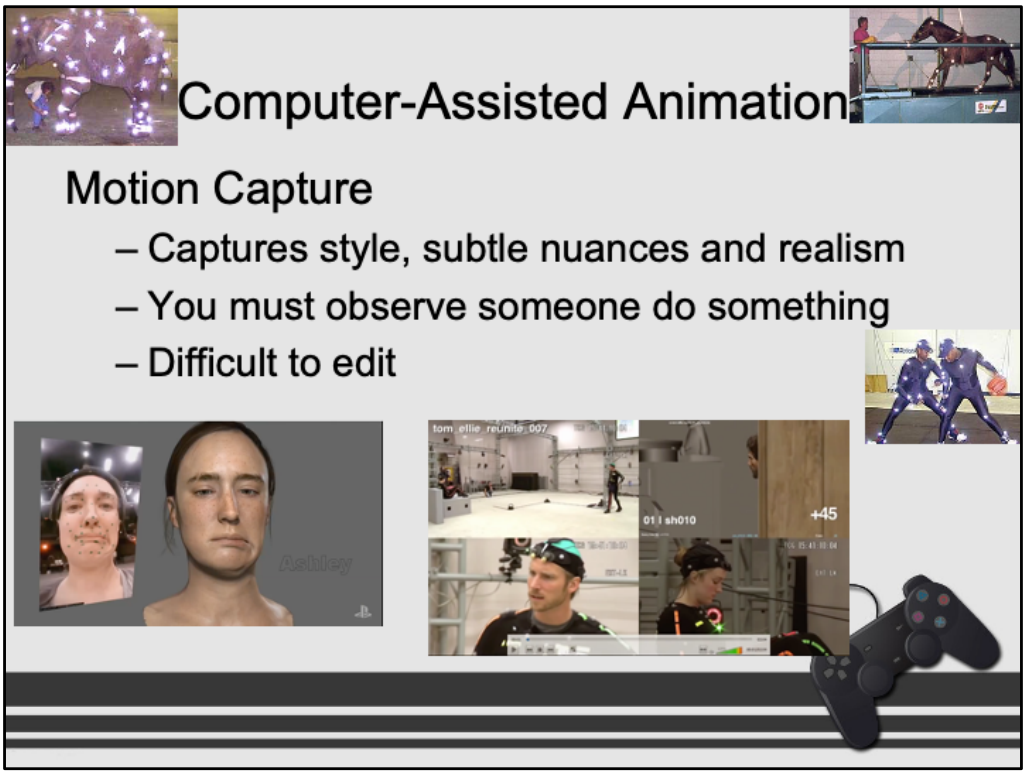
    (masses, forces, inertial properties)
  - Simulate physics by solving equations
  - Realistic but difficult to control



ACM© 1988 "Spacetime Constraints"

Just Cause 3

Embedded video is heavily edited montage by me. Not posted online. Just search for "Just Cause 3" in youtube and watch several to get an idea of the physics

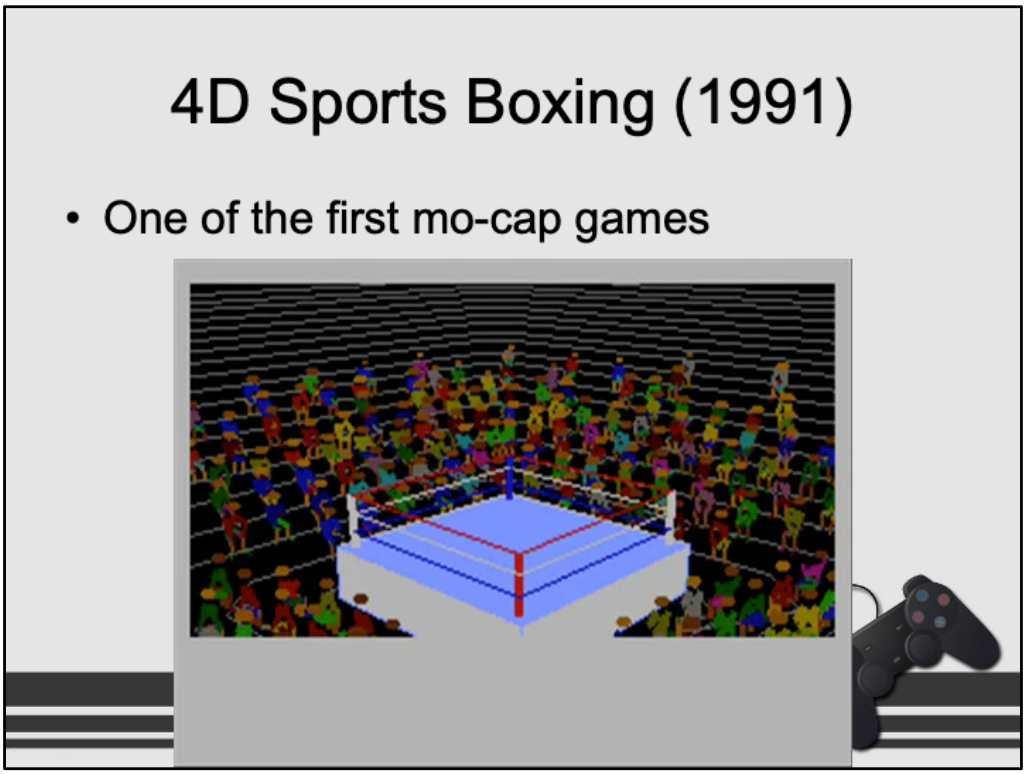


Naughty Dog Mo-cap

https://www.youtube.com/watch?v=mn5_-aZIVlU

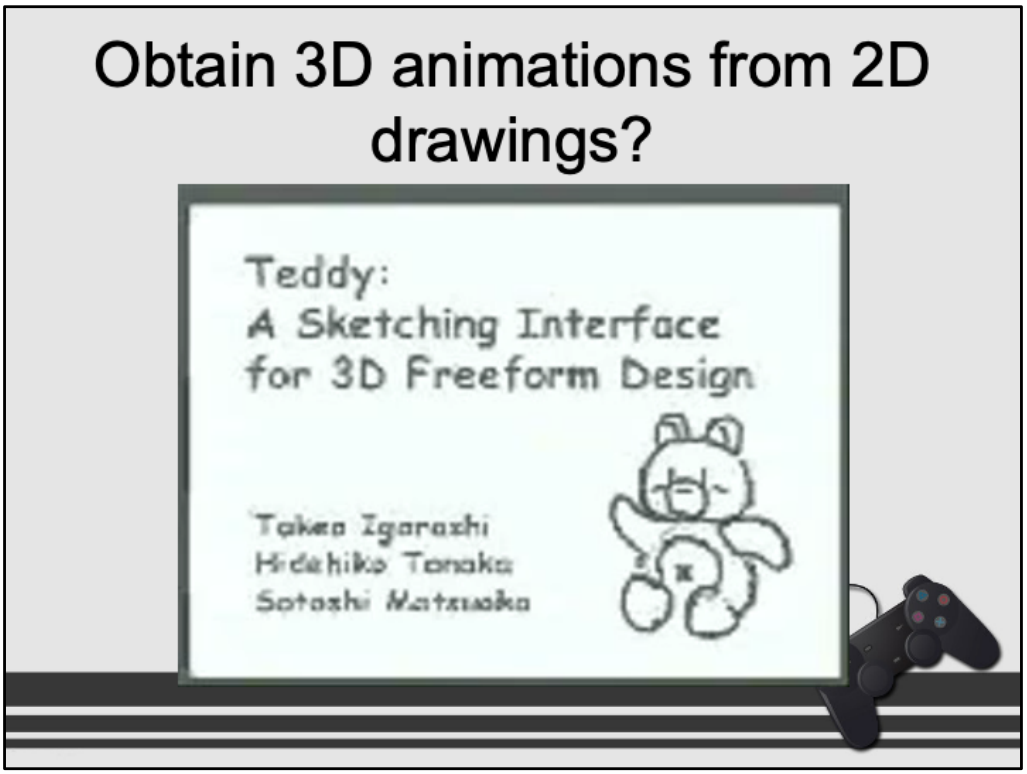
Not same as embed: https://www.youtube.com/watch?v=7_mjLSzW13g

Probably the video I used for Last of Us:
https://www.youtube.com/watch?v=aDVSnaWc5HU



4D sports boxing

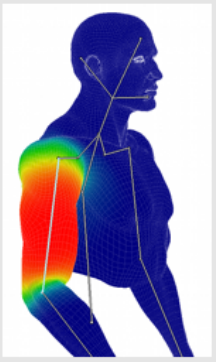
https://www.youtube.com/watch?v=Jt5FrBFXk_8



https://www.youtube.com/watch?v=e2H35SlLmUA

Skeletal Animation

- Abstract skeleton
- Skeleton deforms mesh as it moves
- Keyframes only for skeleton
- Root: 3 or 6DOF; other bones 1/2/3DOF rotation (depending on joint)
- Mesh vertices assigned to bones by weights



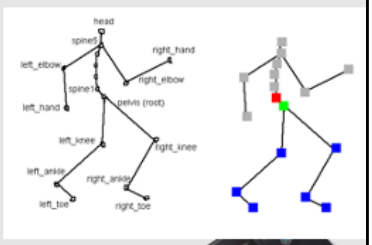


Image from: http://rodolphe-vaillant.fr/?e=29

https://www.youtube.com/watch?v=mhQY2_gVoVg

Image from: http://rodolphe-vaillant.fr/?e=29

Skeletal Animation

- **Advantages**
  - Memory Savings relative to mesh animation
  - Animation blending (multiple animations, and/or IK)
  - Animation portability/reuse
  - Animation authoring is simplified
- **Disadvantages**
  - Implementation difficulty
  - Computational overhead (but can be done in shader!)
  - Problems with realistic mesh deformations (e.g. preserving mesh volume, muscle flex, etc.)

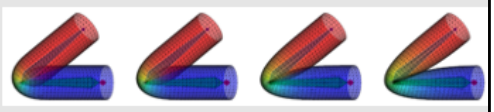


Image from: http://rodolphe-vaillant.fr/?e=29

Half-Life (1998)

- One of the first games with skeletal animation (not just for humanoid!)

https://www.youtube.com/watch?v=rdNB6IDEp-k

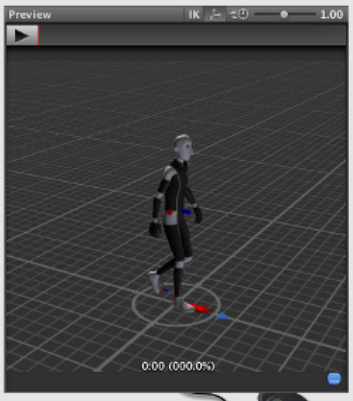
https://www.youtube.com/watch?v=BbP6Jsh8M6Y

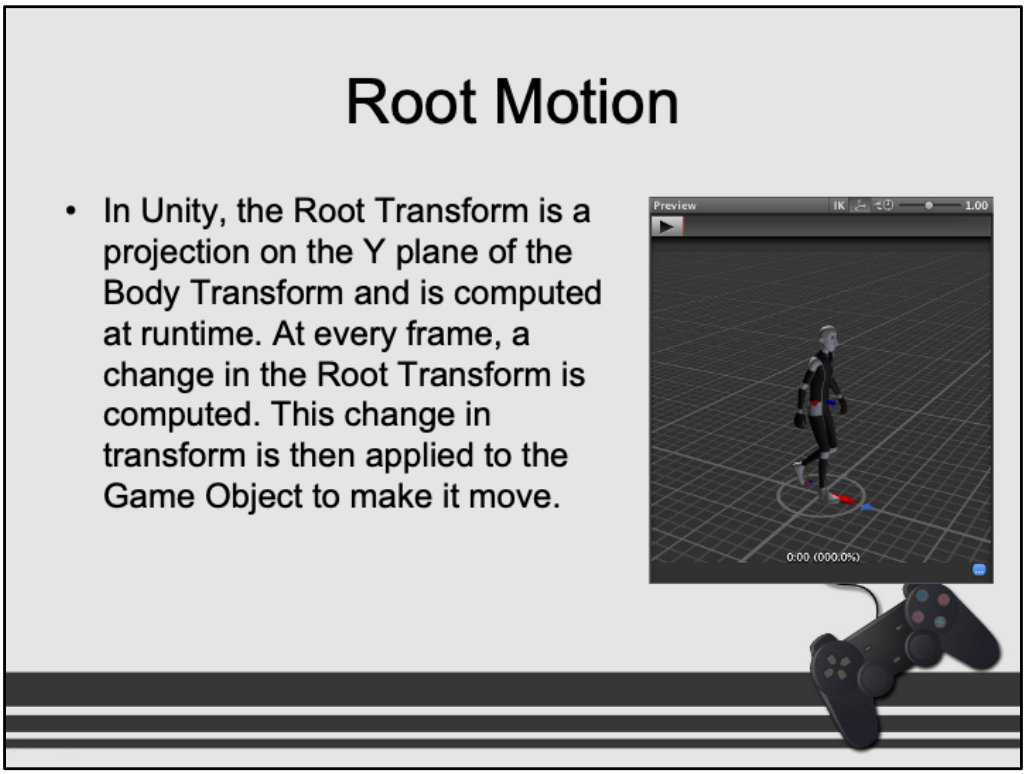
Looks like url changed for some reason

Old:
https://www.youtube.com/watch?v=HYP7rJni1vc

Root Motion

- The root bone (top of bone hierarchy; usually hips/pelvis) has 6DOF
- Root motion refers to the X,Y,Z translation and pitch, yaw, roll rotation of the root bone as a component of a skeletal animation
- This motion can be applied accumulatively to the game object, resulting in locomotion





https://docs.unity3d.com/Manual/RootMotion.html

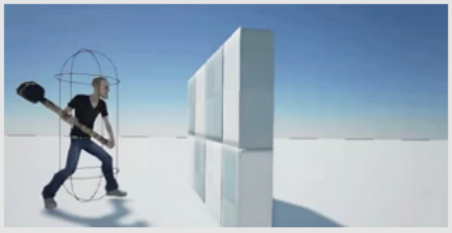
The Body Transform is the mass center of the character. It is used in Mecanim's retargeting engine and provides the most stable displacement model. The Body Orientation is an average of the lower and upper body orientation relative to the Avatar T-Pose.

Generic Root Motion and Loop Pose

This works in essentially the same as Humanoid Root Motion, but instead of using the Body Transform to compute/project a Root Transform, the transform set in **Root Node** (e.g. the hips) is used. The Pose (all the bones which transform below the Root Motion bone) is made relative to the Root Transform

Root Motion - Benefits

- Better cohesion between artist vision and world simulation (avoid capsule separation, movement aligned with foot-fall)
- Declarative (movement spec independent of code)



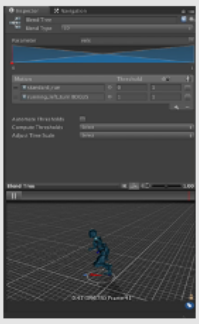
https://docs.unrealengine.com/latest/INT/Engine/Animation/RootMotion/

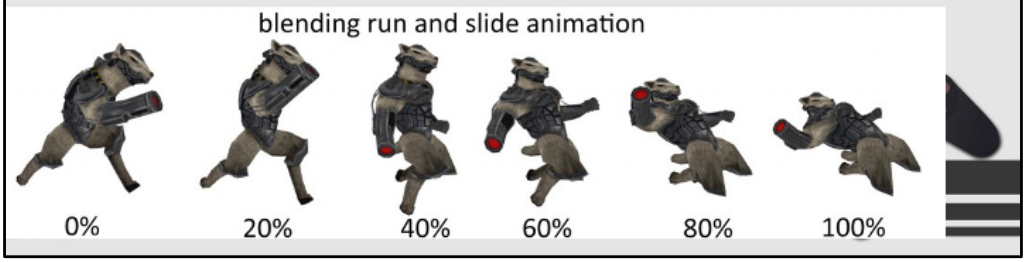
Root Motion Constraints (Unity)

- Limit root transforms to only certain dimensions: rotation, Y-axis (up), XZ
- Green light indicates candidate for bake into pose
- Y-axis: special case. If baked, allows falling according to physics

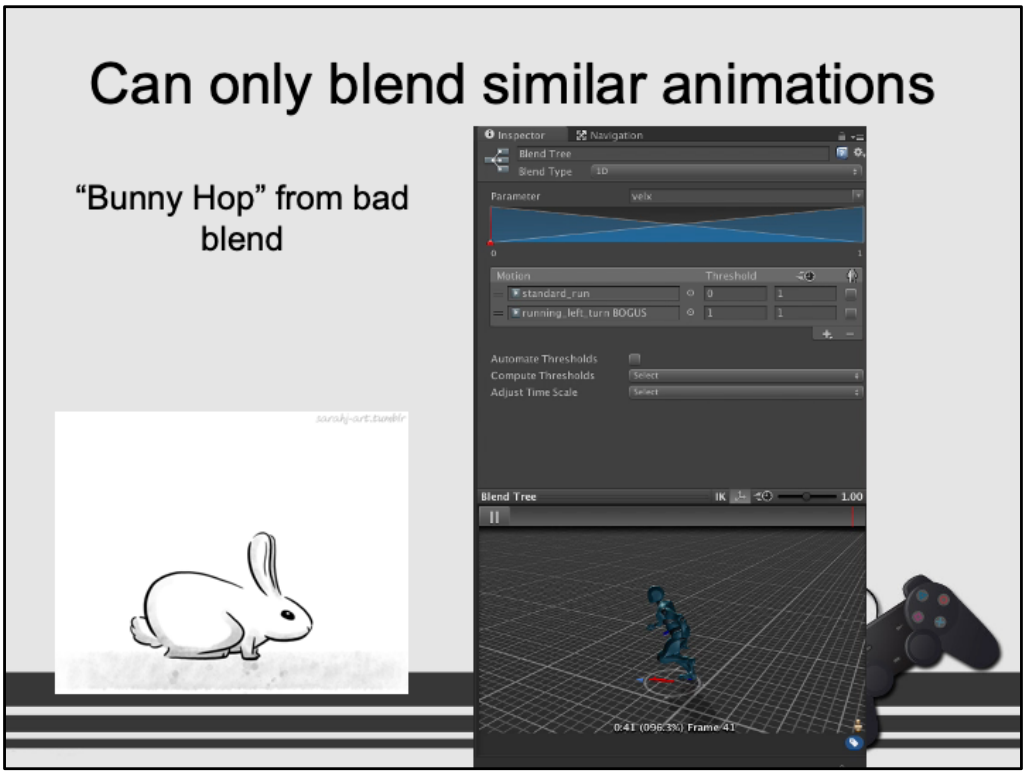
Animation Blending

- Achieve variations between different extremes of similar animations
- **This means just a few animations can define a whole range of movement possibilities!**
- Basic case, usually must be ”similar” animations
- Skeletal animation highly beneficial
- **Root motion can be blended too!**





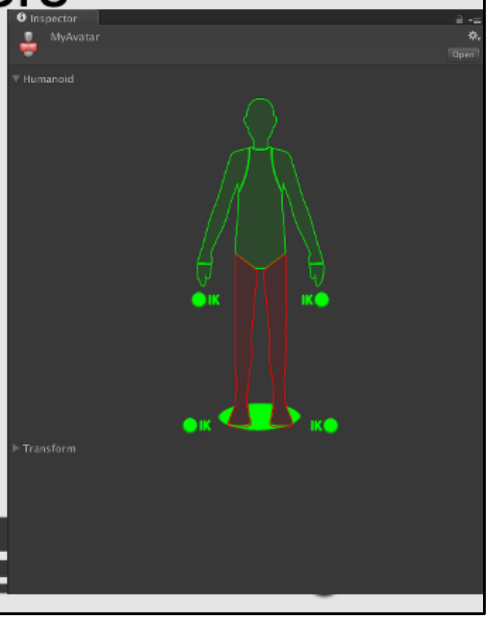
https://markobl.com/2014/12/03/animation-blending-in-monogame-xna/



Bunny Hop animation example – made by me but not hosted

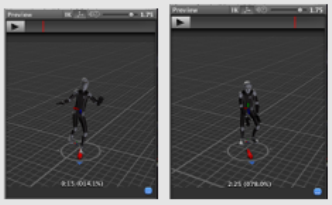
Avatar Masks and Animation Layers

- Layers good for dissimilar animation blending
- Masks allow more control of blending/layering of animations
- Animations that would negatively interfere w/o mask
- Ex: running while shooting
- Works well with IK



Match Targets

- Utility method for Lerp/Slerp with animation timeline and transforms
- Other game engines might not provide or call it something else
- Adjust a limb position/rotation over the course of time to land in a particular pose
- Applications:
  - Stand in a specific spot
  - Jump and land at specific spots
  - Press a button
  - Grab hold of a ledge
- Unity:Animator.MatchTarget()
- Problem: introduces sliding



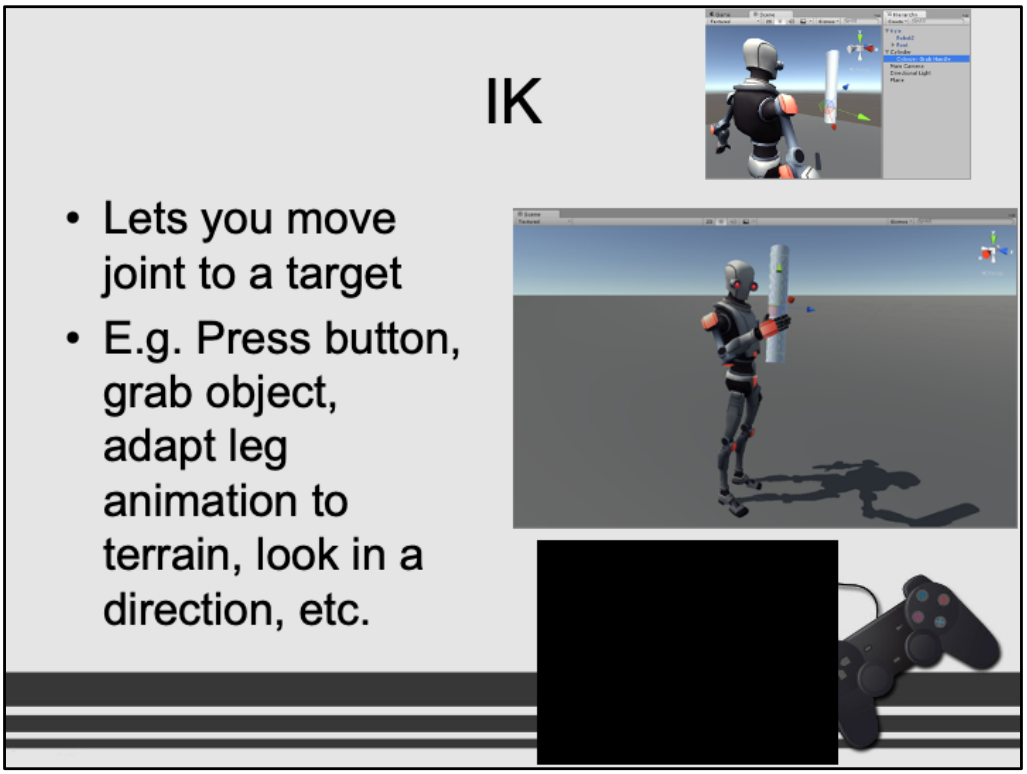
https://docs.unity3d.com/ScriptReference/Animator.MatchTarget.html

https://www.youtube.com/watch?v=JaLh7fkCsKY

Inverse Kinematics

- Forward Kinematics
  - Given the skeleton parameters (position of the root and the joint angles) ***p*** and the position of the sensor/effecter in local coordinates v_s, what is the position of the sensor in the world coordinates v_w.
  - Not too hard, we can solve it by evaluating $\mathbf{S}(\mathbf{p})v_s$
- Inverse Kinematics
  - Given the the position of the sensor/effecter in local coordinates v_s and the position of the sensor in the world coordinates v_w, what are the skeleton parameters ***p***.
  - Much harder requires solving the inverse of $\mathbf{S}(\mathbf{p})$

  the non-linear function
  - We can solve it by root-finding $\mathbf{p}$? such that $\mathbf{S}(\mathbf{p})v_s - \mathrm{v}_w = 0$
  - We can solve it by optimization $\underset{\mathbf{p}}{\text{minimize}} \ \left(\mathbf{S}(\mathbf{p})v_s - \mathrm{v}_w\right)^2$



https://www.youtube.com/watch?v=_W_uE70YuHQ

IK and Authoring

- IK also facilitates manual animation creation (or perhaps puppetry)

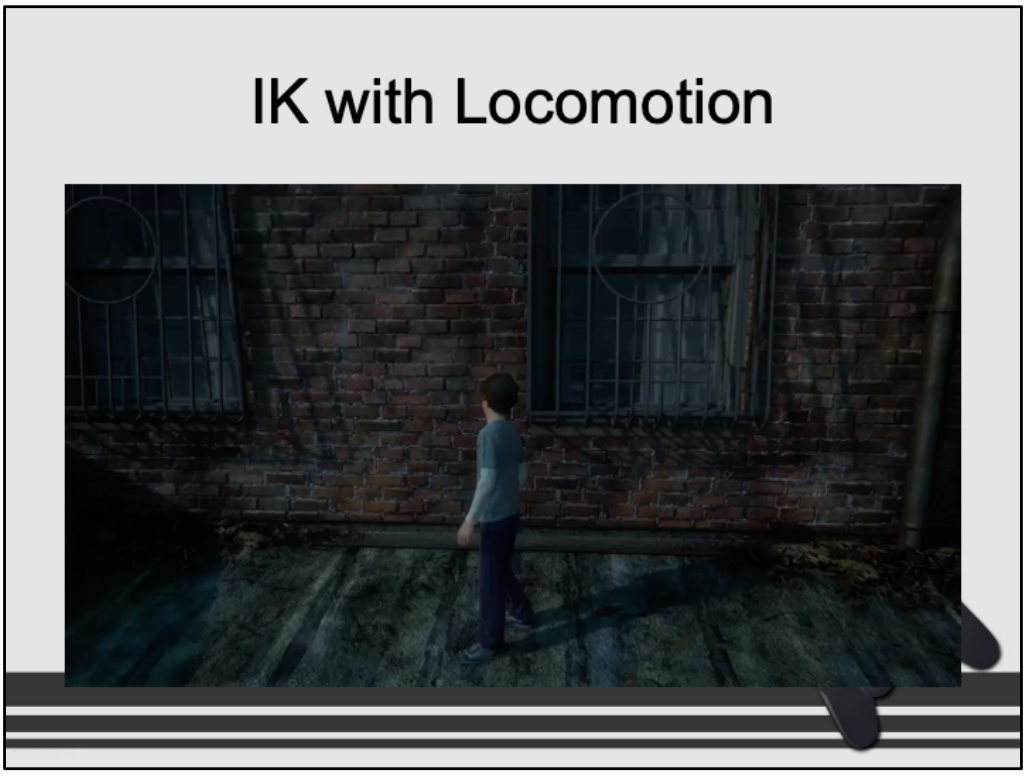
IK in Unity

- public void **SetIKPositionWeight**(AvatarIKGoal **goal**, float **value**)
- public void **SetIKRotationWeight**(AvatarIKGoal **goal**, float **value**)
- public void **SetIKPosition**(AvatarIKGoal **goal**, Vector3 **goalPosition**)
- public void **SetIKRotation**(AvatarIKGoal **goal**, Quaternion **goalRotation**);
- public void **SetLookAtPosition**(Vector3 **lookAtPosition**);

- Weights can be interpolated [0,1], perhaps from an animation's normalized time (for instance, a button press animation)

- AvatarIKGoal: Left/RightFoot, Left/RightHand

AvatarIKGoal refers to an enumeration of skeleton bones



Uncharted 4

Constraint feet, legs
Pivot foot

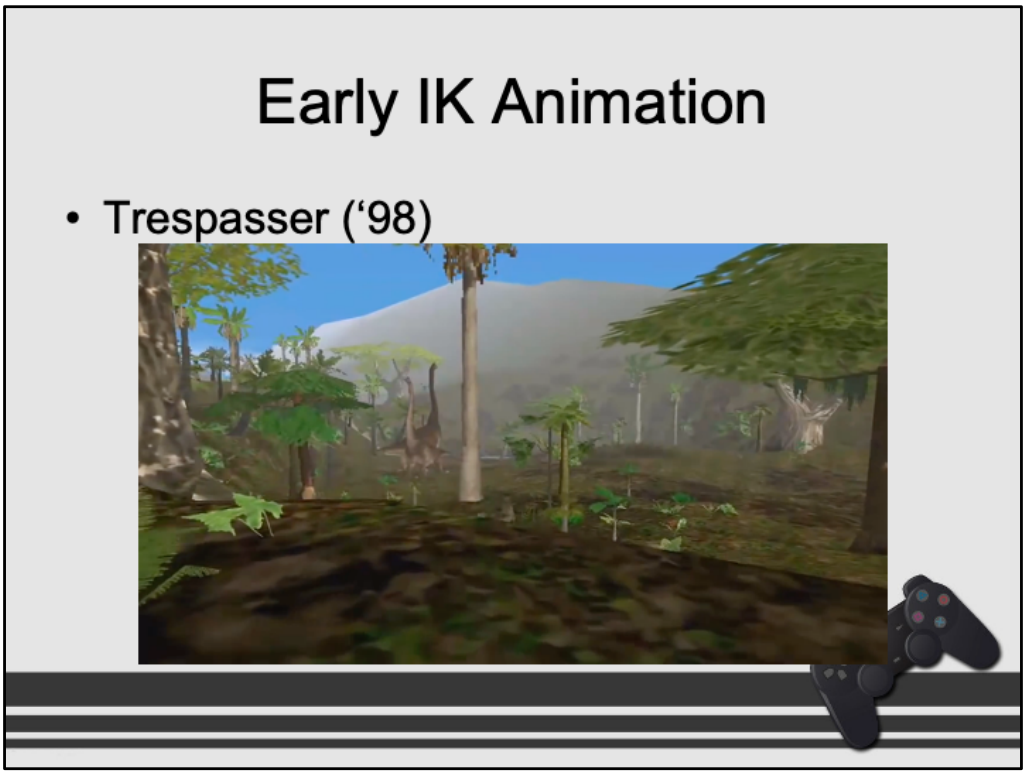
Captured by me – not hosted

Early IK Animation

- Terra Nova Strike Force Centauri (‘96)

Animations with IK adjustments of legs

https://www.youtube.com/watch?v=YdBGaNTBs8U



Animations with IK adjustments of legs

Possibly not the video I embedded, but close. Ignore the commentary

https://www.youtube.com/watch?v=3CMGxd9nhPw&t=1207s

Retargeting Skeletal Animations

- Map animation from one skeleton to another
- Unity's Muscle Space (normalized skeletal movements – rel. to joint limits)
  - Foot IK correction
- Must also worry about interpenetration and joint limits
  - Mixamo Fuse *Arm Spacing* for bulky characters

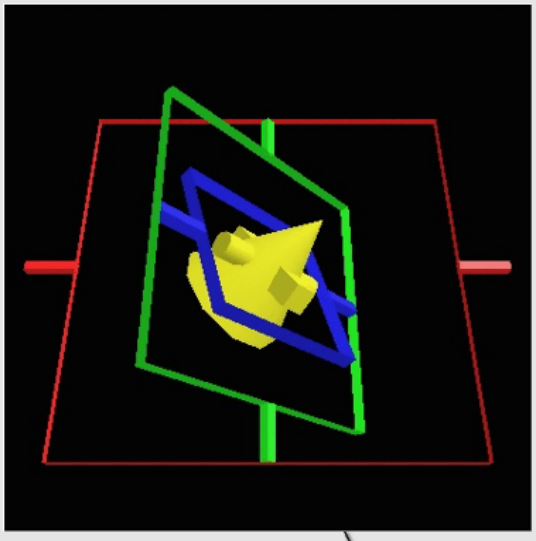
Detailed info on how Unity works

https://blogs.unity3d.com/2014/05/26/mecanim-humanoids/

Rufus from Street Fighter IV

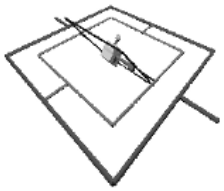
Euler Angles

An euler angle is a rotation about a single axis. Any orientation can be described composing three rotation around each coordinate axis. We can visualize the action of the Euler angles: each loop is a rotation around one coordinate axis.



Interpolating Euler Angles

- **Natural orientation representation**: three angles for three degrees of freedom
- **Unnatural interpolation:** A rotation of 90-degrees first around the z-axis and then around the y-axis has the effect of a 120-degree rotation around the axis (1, 1, 1). But rotation of 30-degrees around the z- and y-axis does not have the effect of a 40-degree rotation around the axis (1, 1, 1).
- **Gimbal lock:** two or more axis align resulting in a loss of rotation degrees of freedom. For example, if the green loop in previous slide aligns with the red loop then both the rotation around the blue loop and the rotation around the red loop produces identical rotation.



Quaternion Interpolation

- **L**inear int**erp**olation (lerp) of quaternion representation of orientations gives us something better:

$$\text{lerp}(\mathbf{q}_0, \mathbf{q}_1, t) = \mathbf{q}(t) = \mathbf{q}_0(1-t) + \mathbf{q}_1 t$$

- Quaternion Refresher $\vec{v} = (\vec{v}_1, \vec{v}_2, \vec{v}_3)$ $\mathbf{q} = (s, \vec{v})$
  - a *general* quaternion **q** consists of four numbers: a scalar *s* and a 3-D vector $\mathbf{q}_1\mathbf{q}_2 = \left(s_1 s_2 - (\vec{v}_1 \bullet \vec{v}_2), s_1\vec{v}_2 + s_2\vec{v}_1 + \vec{v}_1 \times \vec{v}_2\right)$
  - two general quaternions are multiplied by a special rule:
    $\mathbf{q} = (\cos(\theta/2), \sin(\theta/2)\vec{a})$
  - a *unit* quaternion
    can represent a rotation of θ radians around the unit axis vector $\vec{a}$