

## COMP220: Graphics & Simulation

# 7: Rigging and animation

# Learning outcomes

- ▶ **Explain** the role of rigging in 3D animation
- ▶ **Describe** how a rigged model is transformed to produce animation
- ▶ **Use** skeletal animation in your own programs

# Scene graphs



# Coordinate spaces

# Coordinate spaces

Model space

# Coordinate spaces

Model space



Model matrix

# Coordinate spaces

Model space



World space

Model matrix

# Coordinate spaces

Model space



World space

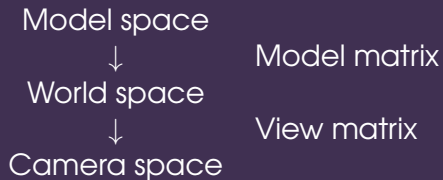


Model matrix

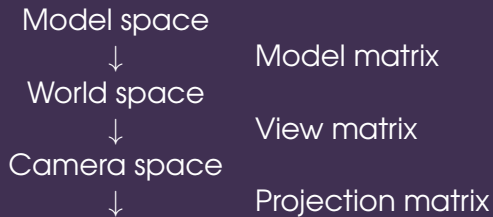
View matrix



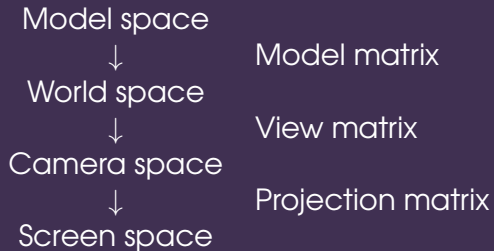
# Coordinate spaces



# Coordinate spaces



# Coordinate spaces



# Rule of thumb

# Rule of thumb

- ▶ When performing calculations, **do not mix** vectors from **different coordinate spaces**

# Rule of thumb

- ▶ When performing calculations, **do not mix** vectors from **different coordinate spaces**
- ▶ E.g. when performing lighting calculations, ensure your fragment position, normal, light direction, eye direction are all in the **same** space

# Scene graph

# Scene graph

- ▶ It is often useful to organise objects into a **hierarchy**



# Scene graph

- ▶ It is often useful to organise objects into a **hierarchy**
- ▶ Each node in the hierarchy has its own model matrix

# Scene graph

- ▶ It is often useful to organise objects into a **hierarchy**
- ▶ Each node in the hierarchy has its own model matrix
- ▶ Transformations stack: object is affected by its own transformation, and that of its parent, and that of its grandparent, and so on

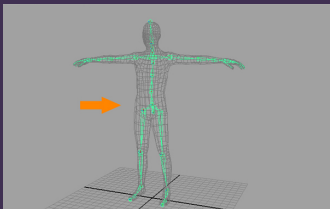
# Scene graph

- ▶ It is often useful to organise objects into a **hierarchy**
- ▶ Each node in the hierarchy has its own model matrix
- ▶ Transformations stack: object is affected by its own transformation, and that of its parent, and that of its grandparent, and so on
- ▶ The model matrix is the **product** of model matrices for the node and its ancestors

# Skeletal animation

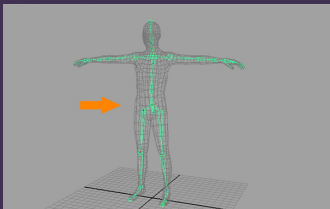


# Rigging



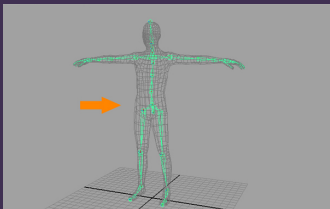
# Rigging

- ▶ A **skeleton** is composed of **bones**

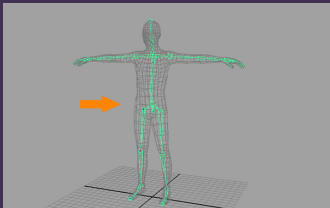


# Rigging

- ▶ A **skeleton** is composed of **bones**
- ▶ Arranged in a **hierarchy**



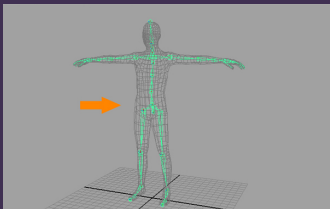
# Rigging



- ▶ A **skeleton** is composed of **bones**
- ▶ Arranged in a **hierarchy**
- ▶ Each bone is essentially just a **transformation**

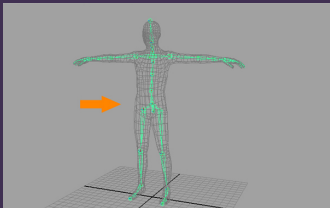


# Rigging



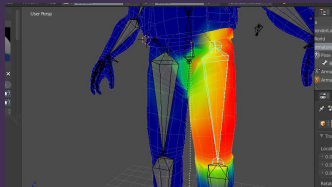
- ▶ A **skeleton** is composed of **bones**
- ▶ Arranged in a **hierarchy**
- ▶ Each bone is essentially just a **transformation**
  - ▶ Usually just rotation around a pivot point

# Rigging



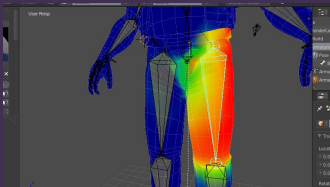
- ▶ A **skeleton** is composed of **bones**
- ▶ Arranged in a **hierarchy**
- ▶ Each bone is essentially just a **transformation**
  - ▶ Usually just rotation around a pivot point
  - ▶ 3D modelling software often represents bones as lines from parent bone to child bone

# Bone weights

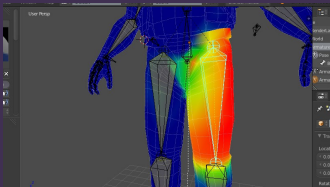


# Bone weights

- Each vertex in the model has a list of **bone weights**

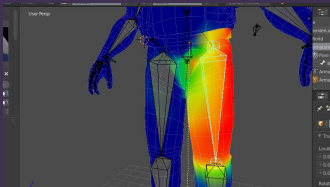


# Bone weights



- ▶ Each vertex in the model has a list of **bone weights**
- ▶ Usually “painted” onto the model by the 3D artist

# Bone weights



- ▶ Each vertex in the model has a list of **bone weights**
- ▶ Usually “painted” onto the model by the 3D artist
- ▶ Weights specify how much each vertex is affected by each bone’s transformation

# Skinning

# Skinning

- ▶ The character is animated by changing the bone transformations



# Skinning

- ▶ The character is animated by changing the bone transformations
- ▶ **Skinning** is the process of applying these transformations to the vertices of the model according to the bone weights

# Skinning

- ▶ The character is animated by changing the bone transformations
- ▶ **Skinning** is the process of applying these transformations to the vertices of the model according to the bone weights
- ▶ Generally handled by a **vertex shader**

# Keyframe animation

# Keyframe animation

- ▶ Most basic form of skeletal animation: specify bone transformations for each frame of animation

# Keyframe animation

- ▶ Most basic form of skeletal animation: specify bone transformations for each frame of animation
- ▶ ... or just for **keyframes** and interpolate between them

# Keyframe animation

- ▶ Most basic form of skeletal animation: specify bone transformations for each frame of animation
- ▶ ... or just for **keyframes** and interpolate between them
- ▶ Keyframes set up by an animator, through motion capture, or a combination of the two

# Keyframe animation

- ▶ Most basic form of skeletal animation: specify bone transformations for each frame of animation
- ▶ ... or just for **keyframes** and interpolate between them
- ▶ Keyframes set up by an animator, through motion capture, or a combination of the two
- ▶ More advanced: can **blend** animations

# Keyframe animation

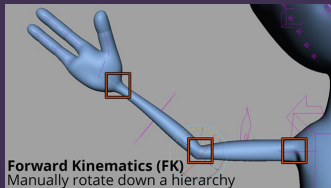
- ▶ Most basic form of skeletal animation: specify bone transformations for each frame of animation
- ▶ ... or just for **keyframes** and interpolate between them
- ▶ Keyframes set up by an animator, through motion capture, or a combination of the two
- ▶ More advanced: can **blend** animations
  - ▶ E.g. blend between walking and running



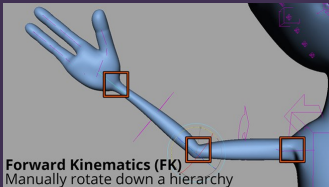
# Keyframe animation

- ▶ Most basic form of skeletal animation: specify bone transformations for each frame of animation
- ▶ ... or just for **keyframes** and interpolate between them
- ▶ Keyframes set up by an animator, through motion capture, or a combination of the two
- ▶ More advanced: can **blend** animations
  - ▶ E.g. blend between walking and running
  - ▶ E.g. bottom half plays “walk” animation, top half plays “fire weapon” animation

# Forward kinematics (FK)

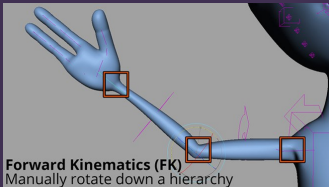


# Forward kinematics (FK)



- Bone transformations are set **explicitly**

# Forward kinematics (FK)

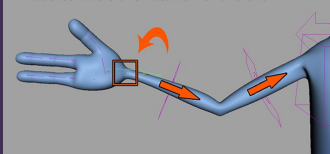


- ▶ Bone transformations are set **explicitly**
- ▶ Children are affected by parent transformations, e.g. if upper arm rotates, lower arm rotates with it

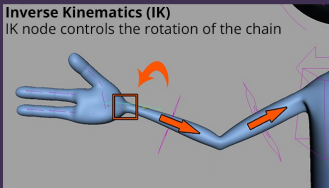
# Inverse kinematics (IK)

## Inverse Kinematics (IK)

IK node controls the rotation of the chain

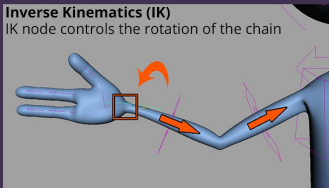


# Inverse kinematics (IK)



- Bone transformations are calculated to reach a **target**

# Inverse kinematics (IK)



- ▶ Bone transformations are calculated to reach a **target**
- ▶ E.g. we want character's hand to touch an object; IK calculates rotations of upper and lower arm to achieve this subject to constraints

# The most common use for IK



# The most common use for IK



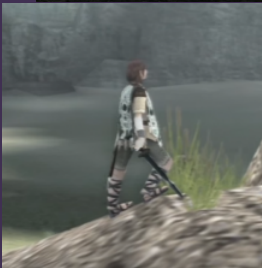
# The most common use for IK



# The most common use for IK



# The most common use for IK



# Ragdolls



# Ragdolls



- Attach a **rigid body** to each bone and run a **physics simulation**

# Ragdolls



- ▶ Attach a **rigid body** to each bone and run a **physics simulation**
- ▶ Often used for death animations

# Procedural animation



# Procedural animation

- ▶ Many games mix some or all of keyframe animation, IK, and physics simulation

# Procedural animation

- ▶ Many games mix some or all of keyframe animation, IK, and physics simulation
- ▶ <https://www.youtube.com/watch?v=JEzxDbIK2Yk>

# Procedural animation

- ▶ Many games mix some or all of keyframe animation, IK, and physics simulation
- ▶ <https://www.youtube.com/watch?v=JEzxDbIK2Yk>
- ▶ <https://www.youtube.com/watch?v=UQdkkmP7amI>

# Advanced animation



# Animating with Math

<http://www.gdcvault.com.ezproxy.falmouth.ac.uk/play/1023249/Animating-With>