Tenghao Wang

Senior Character Technical Artist at Santa Monica Studio

Santa Monica Studio

Graduated from Carnegie Mellon University,
Entertainment Technology Center

Joined Activision Blizzard,
Worked on Infinity Warfare

Joined Visual Concept,
Worked on 2K18, 2K19 and 2K20


Santa Monica Studio™


GOD OF WAR™
RAGNARÖK

Joint-Based Skin Deformation in God of War Ragnarök

Tenghao Wang

Senior Character Technical Artist

Santa Monica Studio

Talk Outline

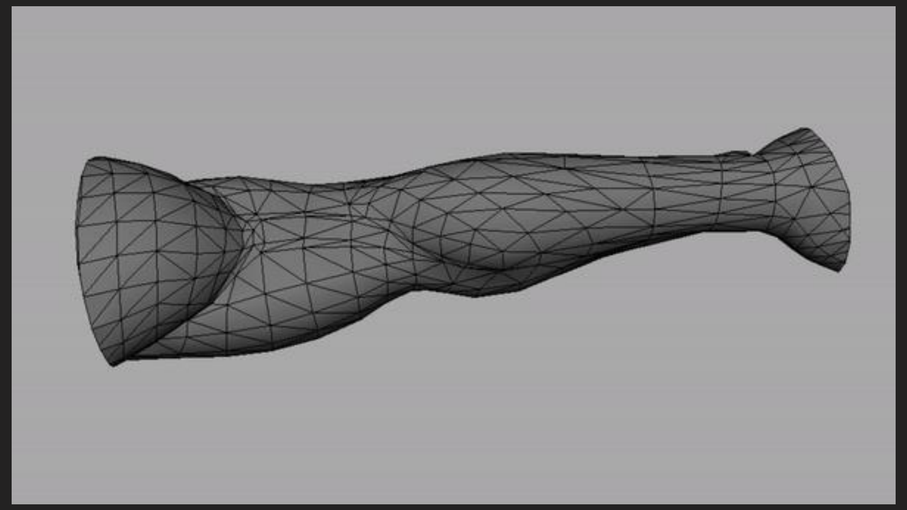
- Joint-Based Muscle Rig
- Joint Dynamics

Santa Monica Studio

Joint Based Muscle Deformation

- Helper Joints

  GDC 2005: Helper Joints: Advanced Deformations on Runtime Characters by Jason Parks



Corrective Joint-Based Rigs By Kiel Friggins

- Skinning Decomposition and RBF Driver

  Dem-Bones or Skinning Converter in Houdini
  RBF Solver

Automatic Creation of Helper Joints with Skinning Decomposition and RBF
By Chad Vernon

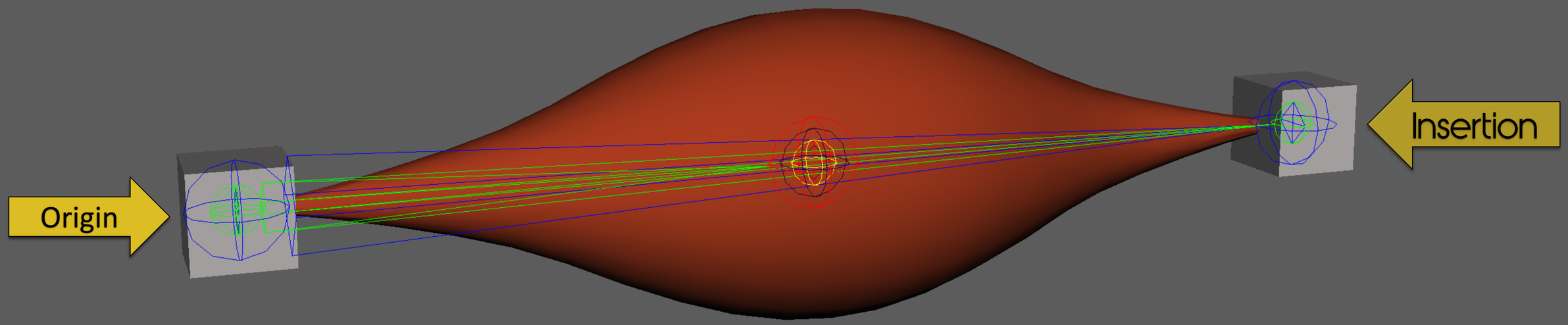
Muscle Simulation with ZivaRT

© 2023 MARVEL

Muscle Simulation with PSD

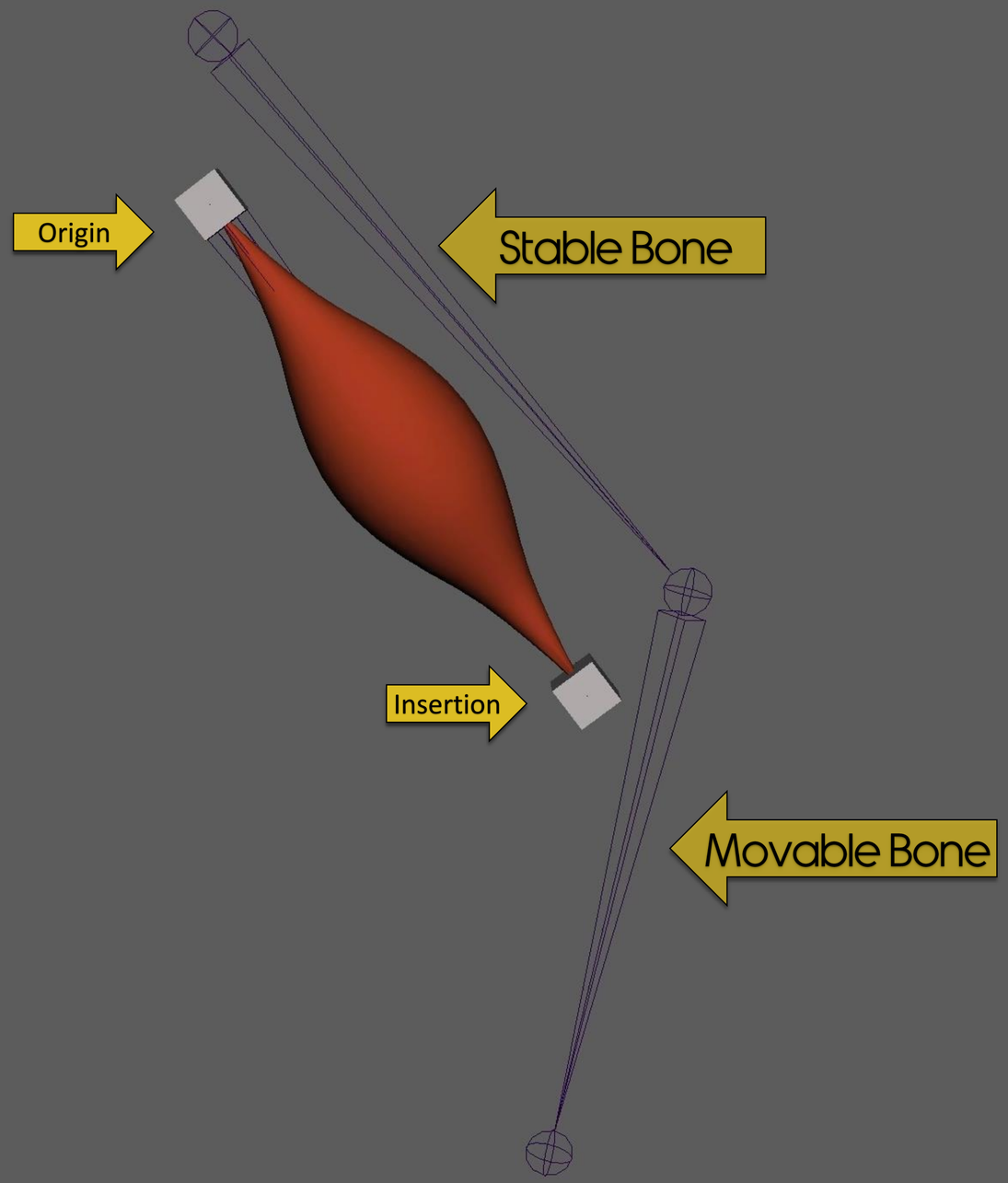

Joint-Based Muscle Rig
NOT ACTUAL SCALES

Muscle Joint Group



Muscle Properties

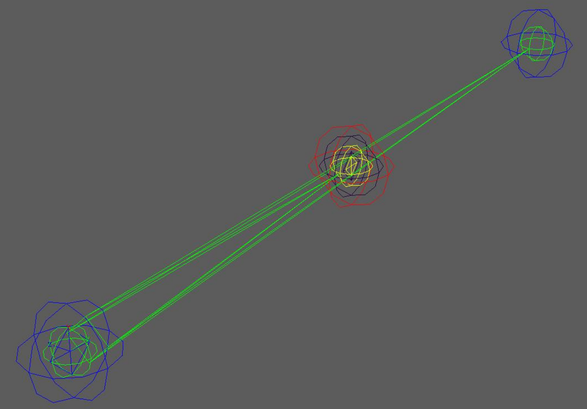
- Curve fiber structure, start - end
- Volume preservation
- Dynamics

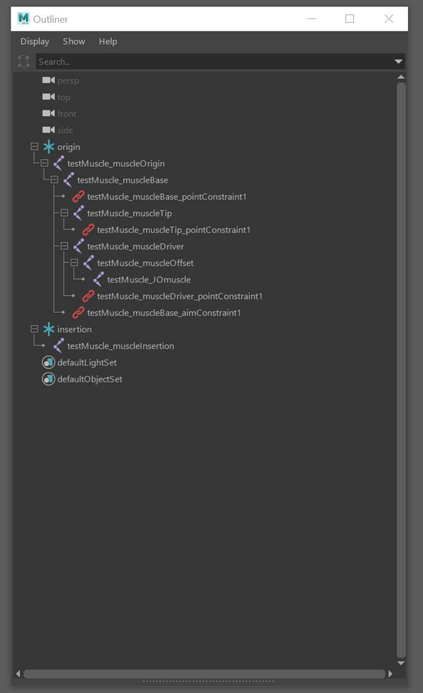


Muscle Joint Group

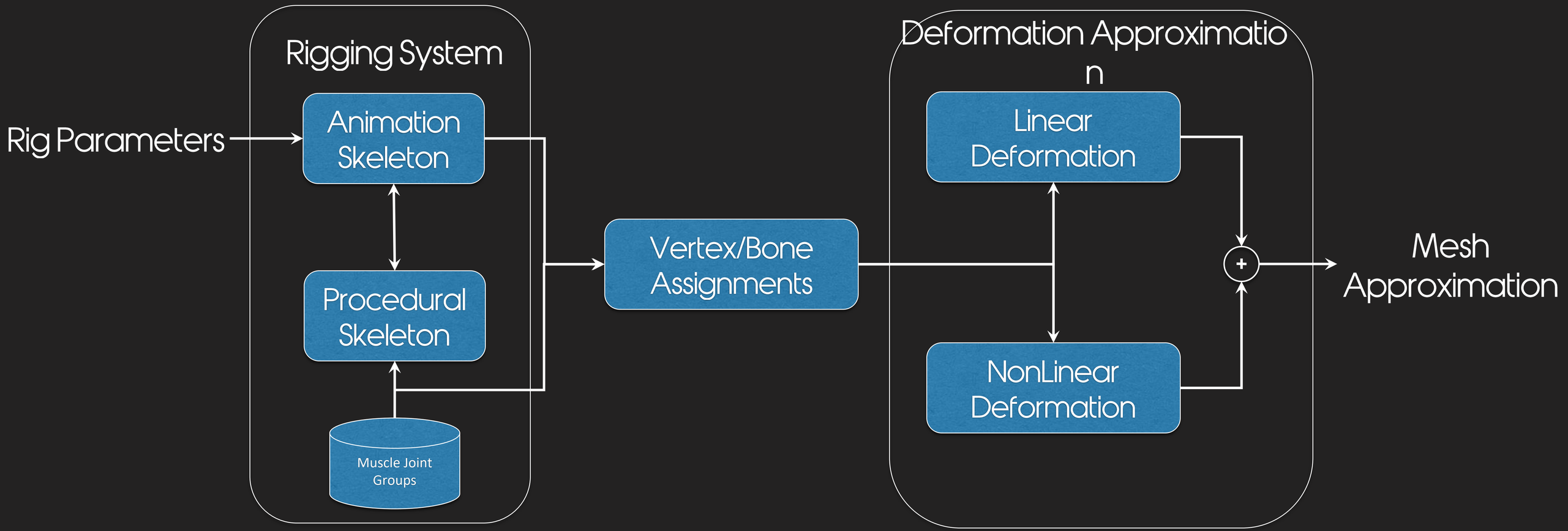
- Use aim constraints to simulate muscle attachments
- Use point constraints to simulate muscl e movement
- Use set driven key to simulate muscl e bulges and correct deformation
- Constraints and SDK are supported in-eng ine

muscle origin and muscle insertion
muscle base and muscle tip
muscle driver
muscle skinned joint





Overview of the Joint-Based Muscle Rig



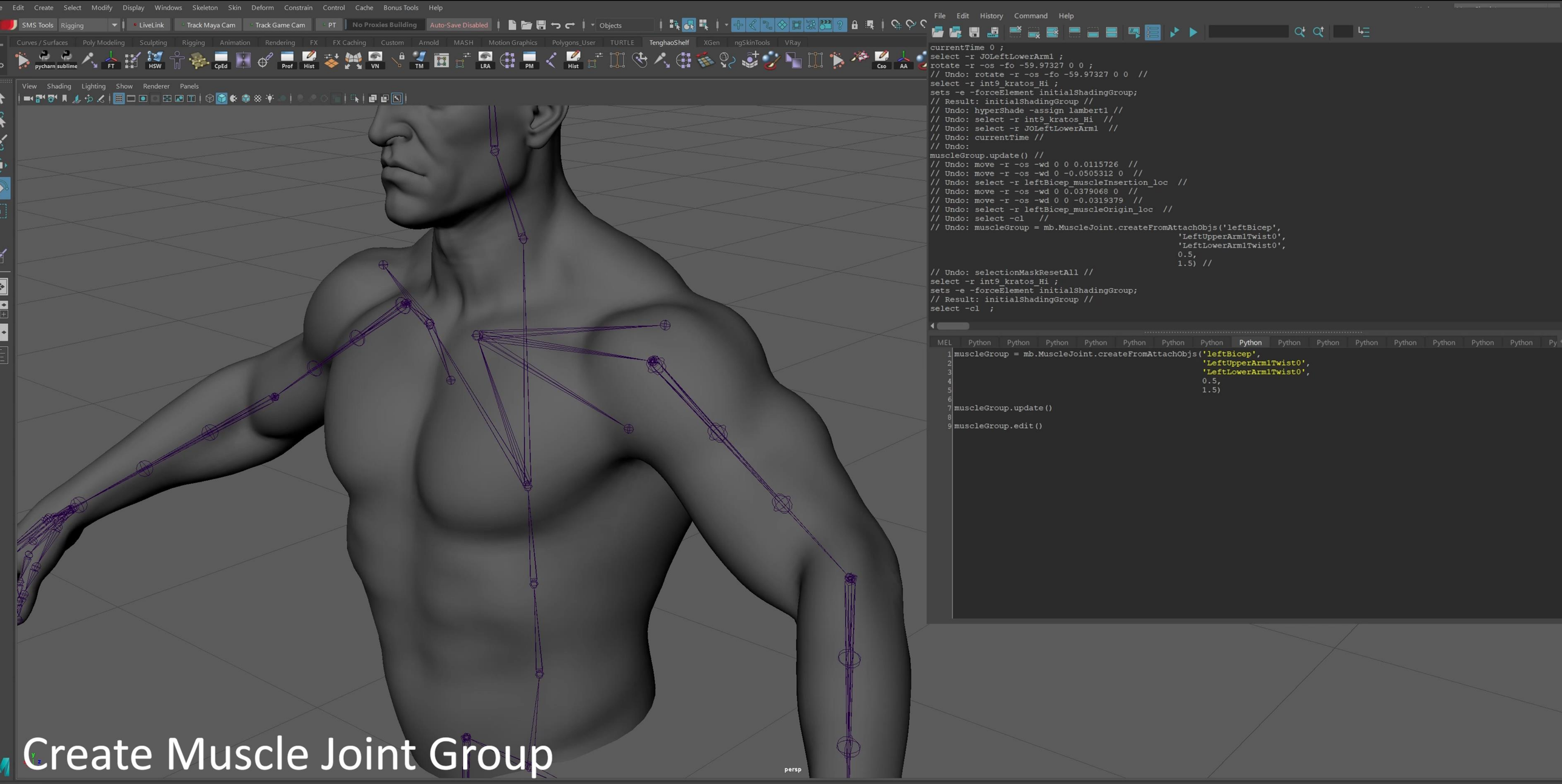
Biped Muscle Template

Muscle Joint Group

```python
@classmethod
def createFromAttachObjs(cls, muscleName, originAttachObj, insertionAttachObj,
                         compressionFactor=1.0,
                         stretchFactor=1.0,
                         stretchOffset=None,
                         compressionOffset=None):
    """

    Args:
        muscleName (str): the name of the muscle
        originAttachObj (str): the name of the immobile bone
        insertionAttachObj (str): the name of the movable bone
        compressionFactor (float): ratio of muscle compression
        stretchFactor (float): ratio of muscle stretch
        stretchOffset list(float, float, float):
                      offset vector when muscle group is stretched
        compressionOffset list(float, float, float):
                      offset vector when muscle group is compressed

    """
def edit(self):
    """Enter the edit mode"""
def update(self):
    """Apply the edits"""
if __name__ == '__main__':
    muscleGroup = MuscleJoint.createFromAttachObjs('leftBicep',
                                                   'LeftUpperArm1Twist0',
                                                   'LeftLowerArm1Twist0',
                                                   compressionFactor=0.5,
                                                   stretchFactor=1.5)
    # apply edits to muscle group
    muscleGroup.update()
    # enter edit mode
    muscleGroup.edit()
```

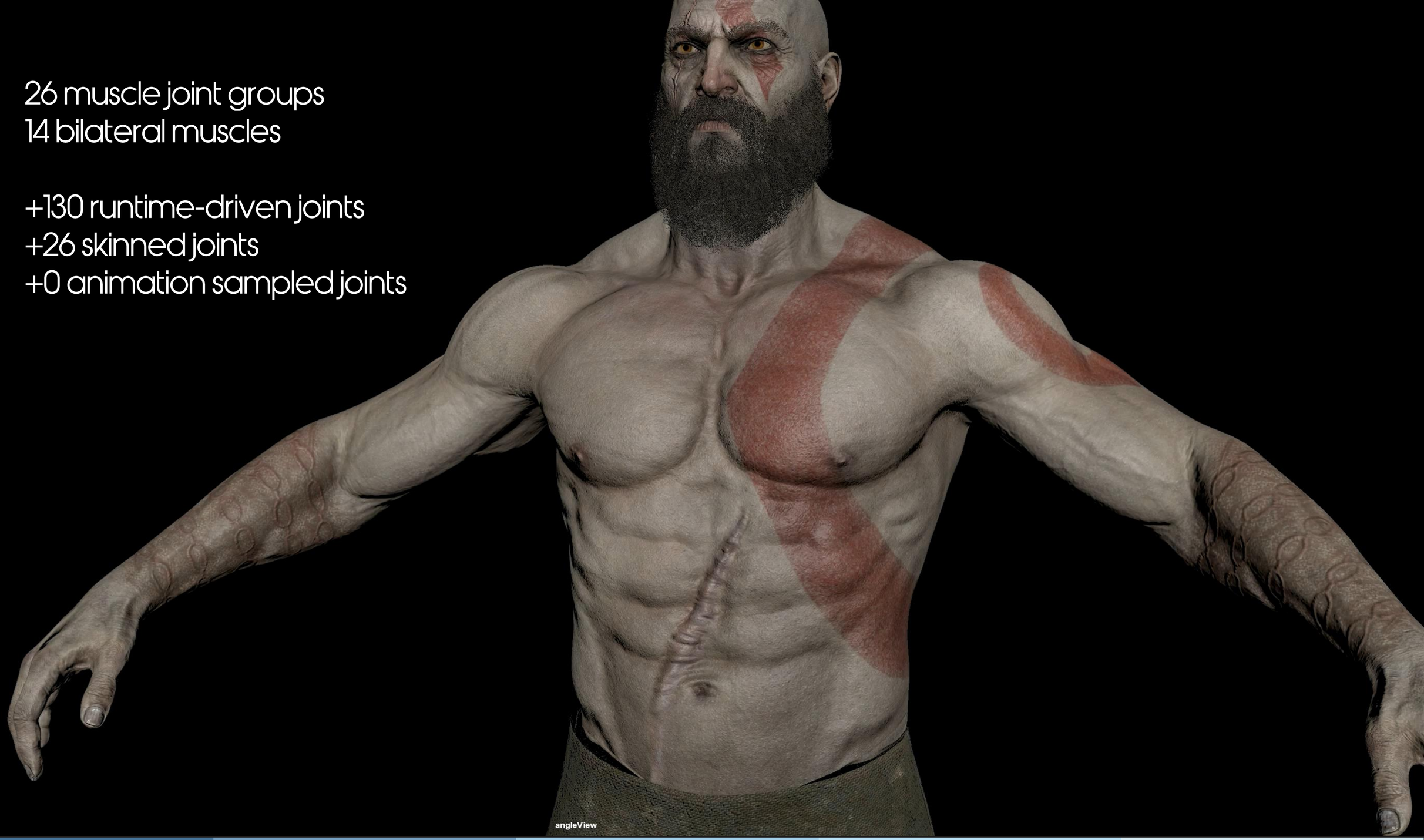
Muscle Component

```python
class UpperArmMuscles(BipedMuscles):
    def __init__(self, name='upperArmMuscles', side='Left',
                 upperArmJoint=None, lowerArmJoint=None,
                 upperArmTwistJoint=None, lowerArmTwistJoint=None,
                 scapulaJoint=None, inferiorAngleJoint=None):
        super(UpperArmMuscles, self).__init__(name=name, side=side)


    def add(self):

        self.bicep = MuscleJoint.createFromAttachObjs(muscleName='{0}Bicep'.format(self.side),
                                                      originAttachObj=self.upperArmTwistJoint,
                                                      insertionAttachObj=self.lowerArmTwistJoint,
                                                      compressionFactor=0.5,
                                                      stretchFactor=1.5,
                                                      bulgeVector=[0.0, 0.0])



        self.tricep = MuscleJoint.createFromAttachObjs(muscleName='{0}Tricep'.format(self.side),
                                                       originAttachObj=self.scapulaJoint,
                                                       insertionAttachObj=self.lowerArmJoint,
                                                       compressionFactor=0.5,
                                                       stretchFactor=1.5,
                                                       bulgeVector=[0.0, 0.0])
```


Create Muscle Joint Group

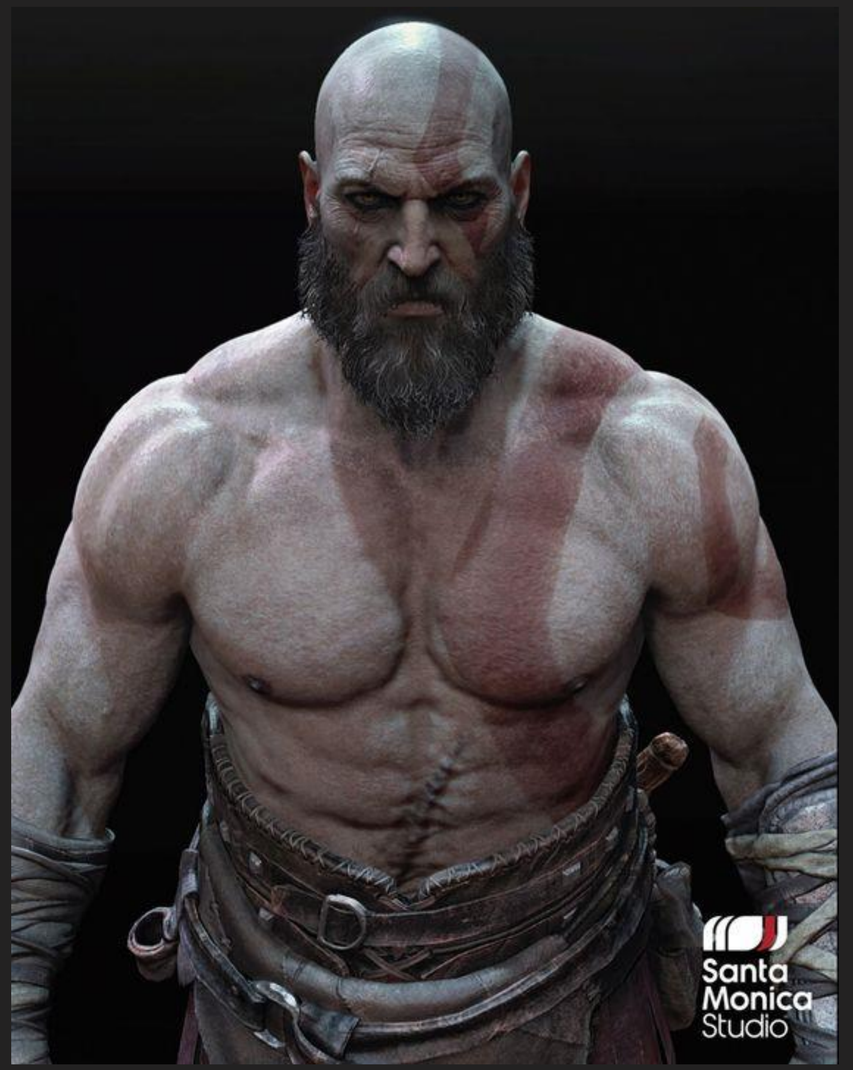
- 26 muscle joint groups
- 14 bilateral muscles

- +130 runtime-driven joints
- +26 skinned joints
- +0 animation sampled joints



Advantages of the Joint-Based Muscle Rig

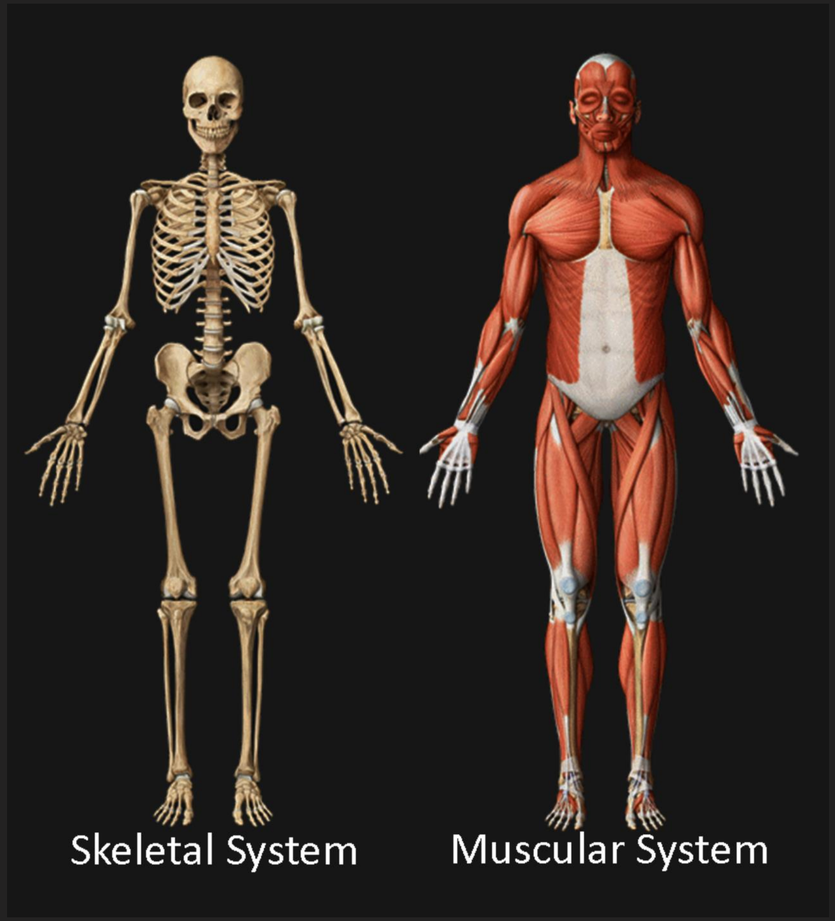
- No highly realistic blendshapes needed
- Not depend on registered pose spaces
- Compatible with the game engine
- Speed up rig evaluation and save memory usage
- A generic and systematic rigging solution



Implementation Details

Musculoskeletal System

- Provides our body with movement, stability, shape, and support
- Subdivided into two systems: skeletal system and muscular syste m
- Skeletal muscles are located between bones with origin and inser tion
- Skeletal muscles are the ones that act on the body joints to produ ce movements



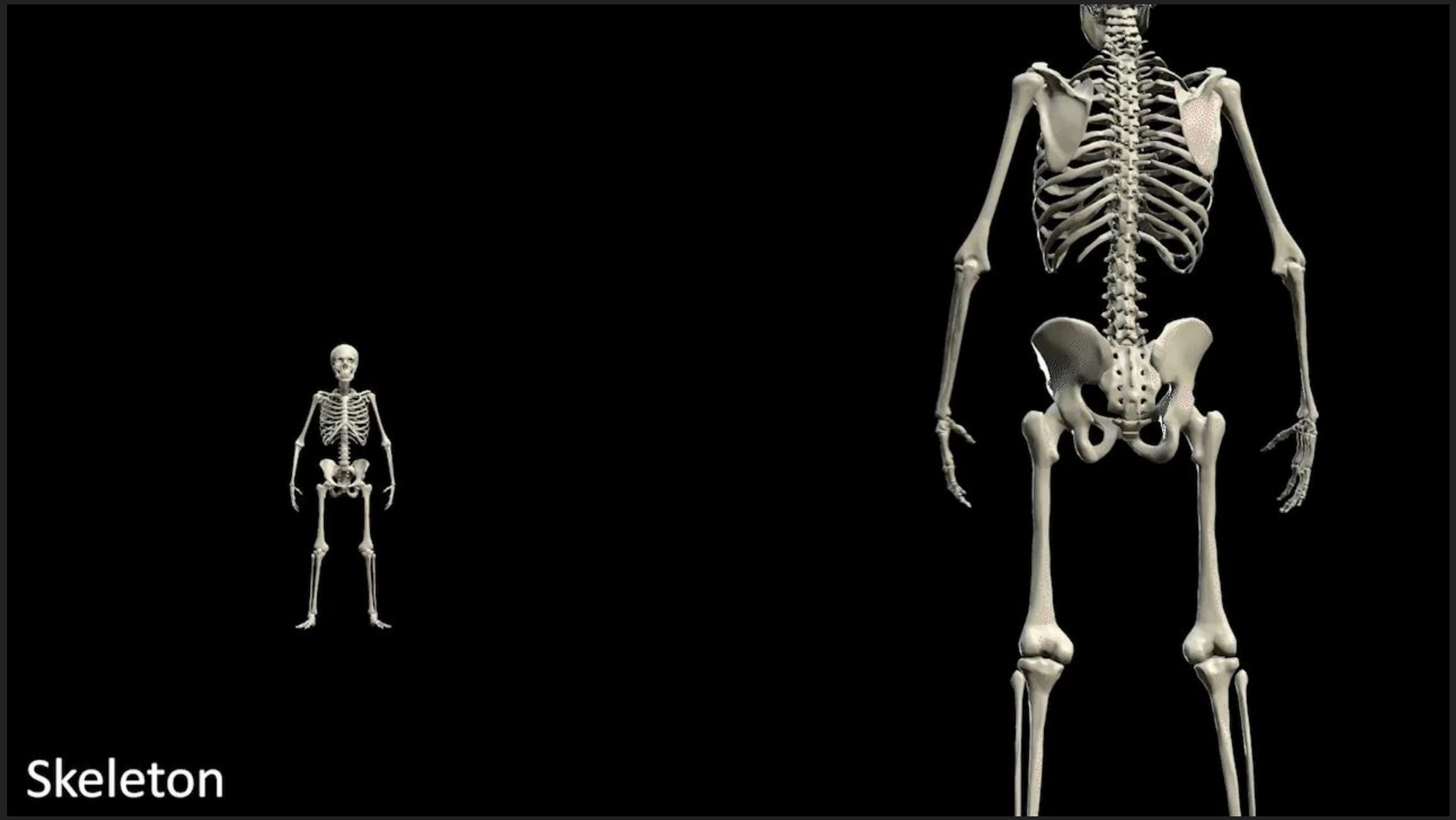
Skeletal System

- Animation Skeleton
- Procedural Helper Joints

Animation Skeleton

- Core skeleton that is necessary to describe t he motion accurately.
- Standard and fixed for production
- A shared asset between different departm ents
- Can be broken down into four major areas: t he spine, skull, arms and legs

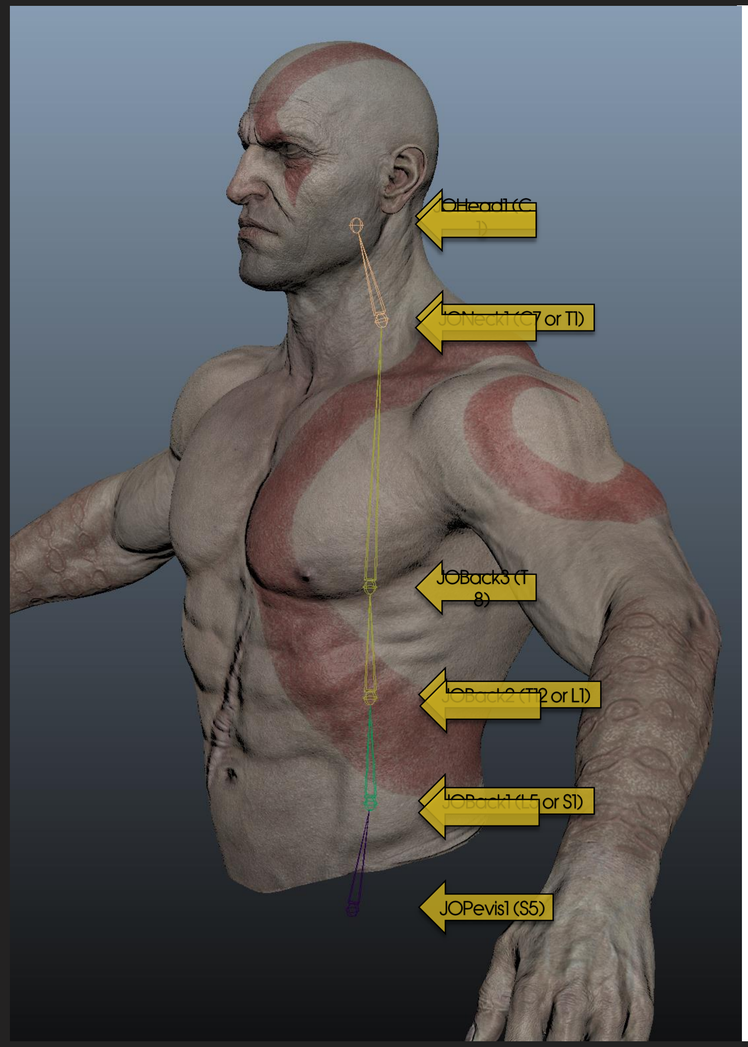
Animation Skeleton Used in God of War: Ragnarok

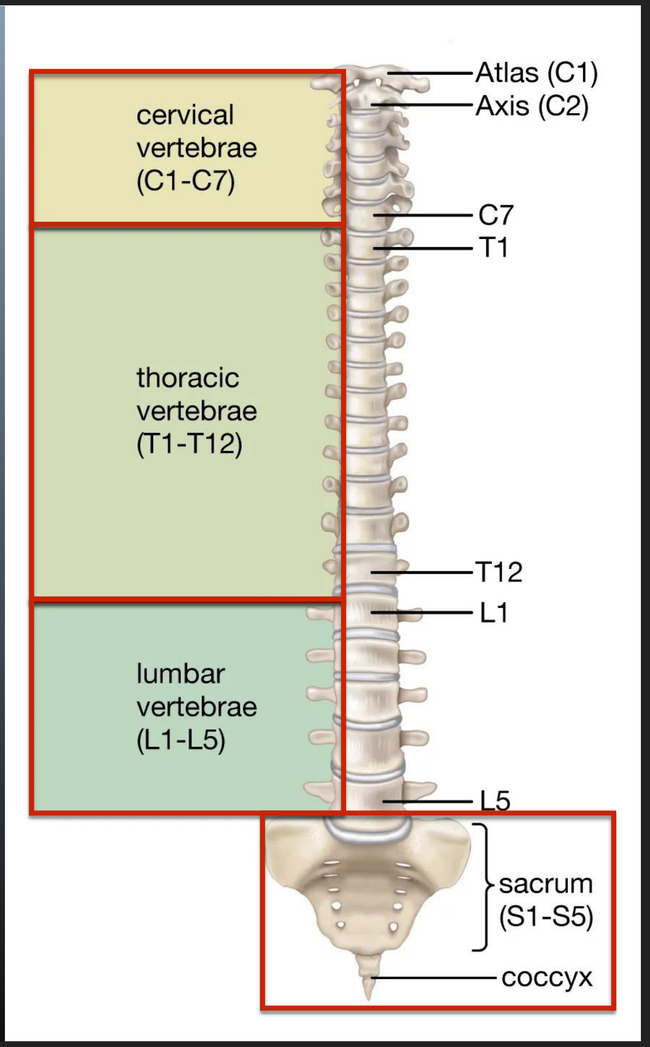


An example of muscle simulation in Ziva VFX

Spinal Column

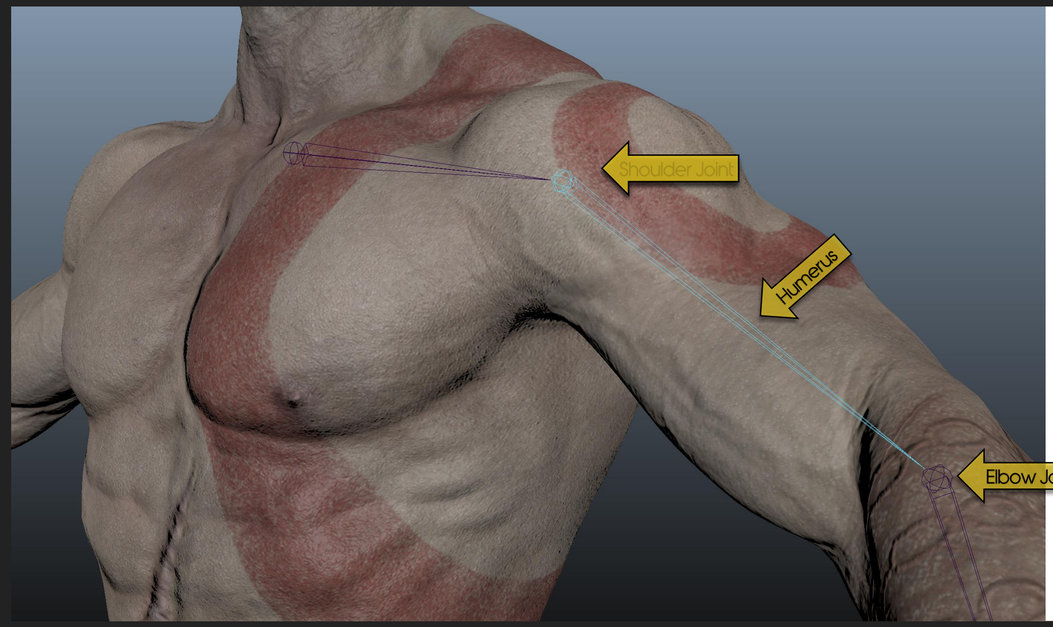
- JOHead1 maps to 1st cervical vertebra (C1)
- JONeck1 maps to 7th cervical vertebra (C7)
- JOBack3 maps to 8th thoracic vertebra (T8)
- JOBack2 maps to 12th thoracic vertebra (T12)
- JOBack1 maps to 5th lumbar vertebra (L5)
- JOPelvis1 maps to 5th sacral vertebra (S5)

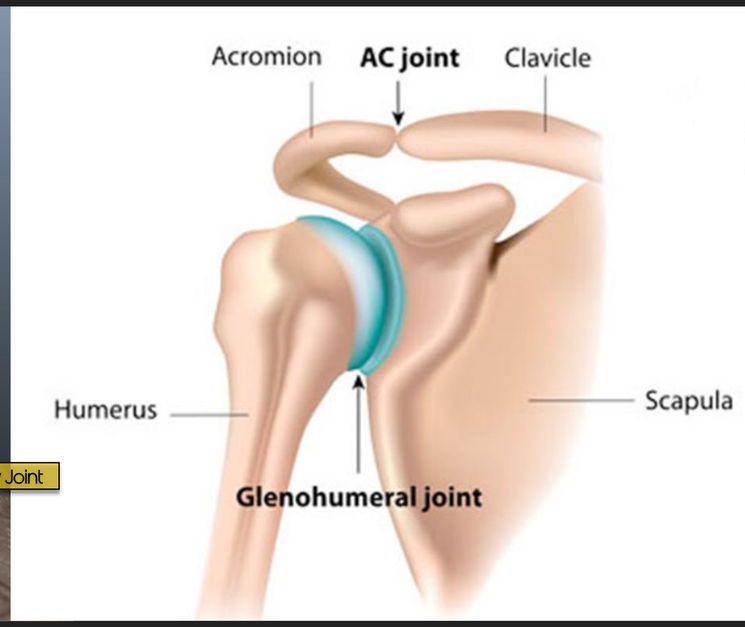


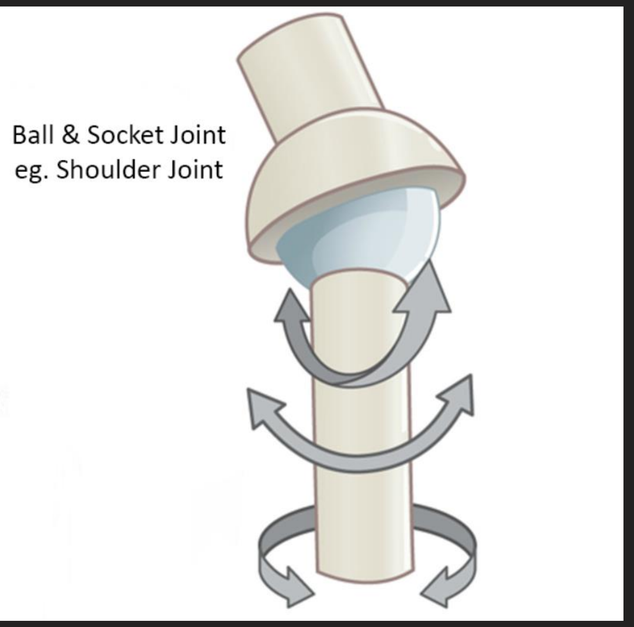


Humerus (Upper Arm Bone)

- Runs from the shoulder and scapula to the elbow
- Ball-and-socket joint allows for movement with 3 degrees of freedom

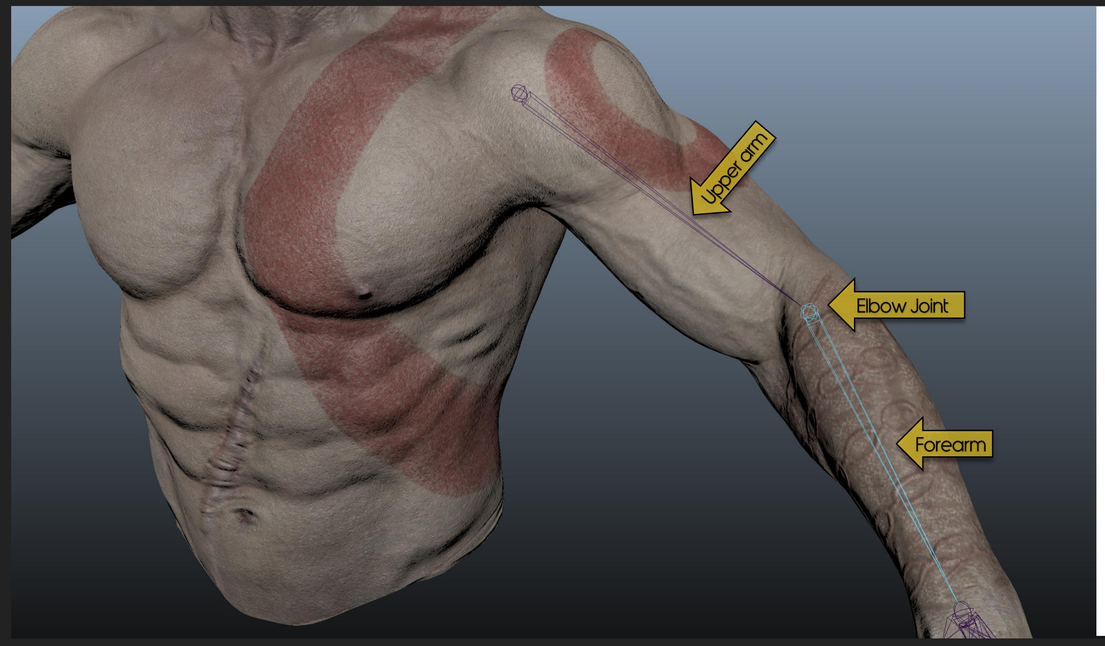


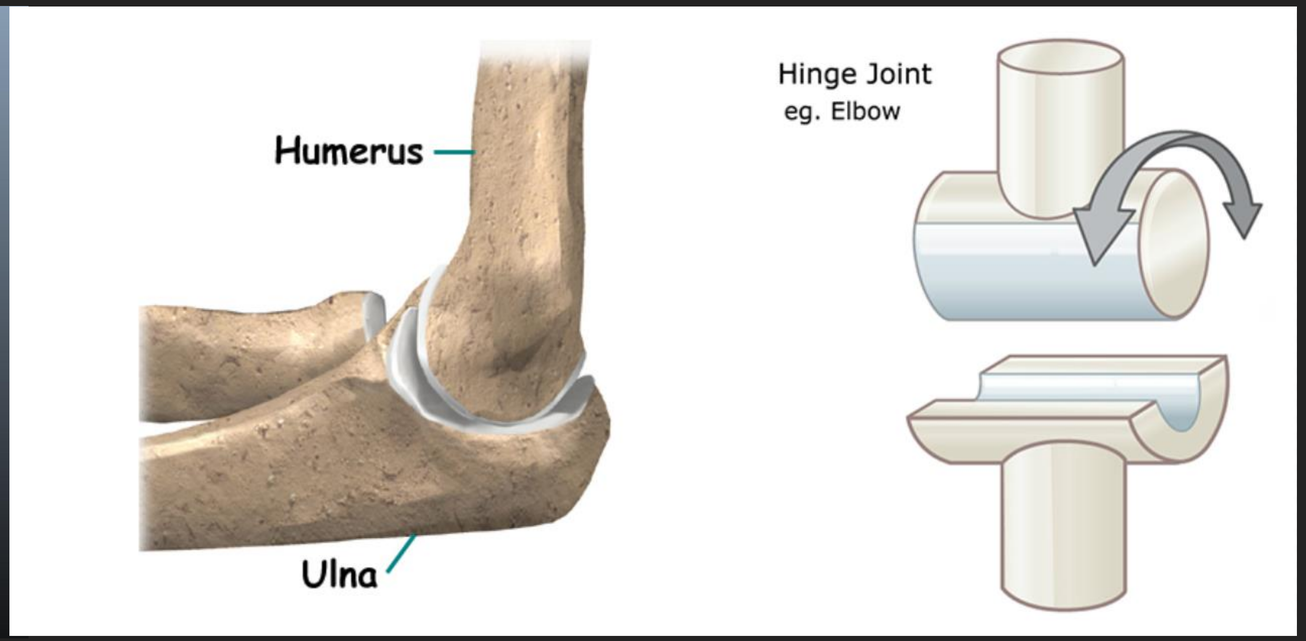




Elbow

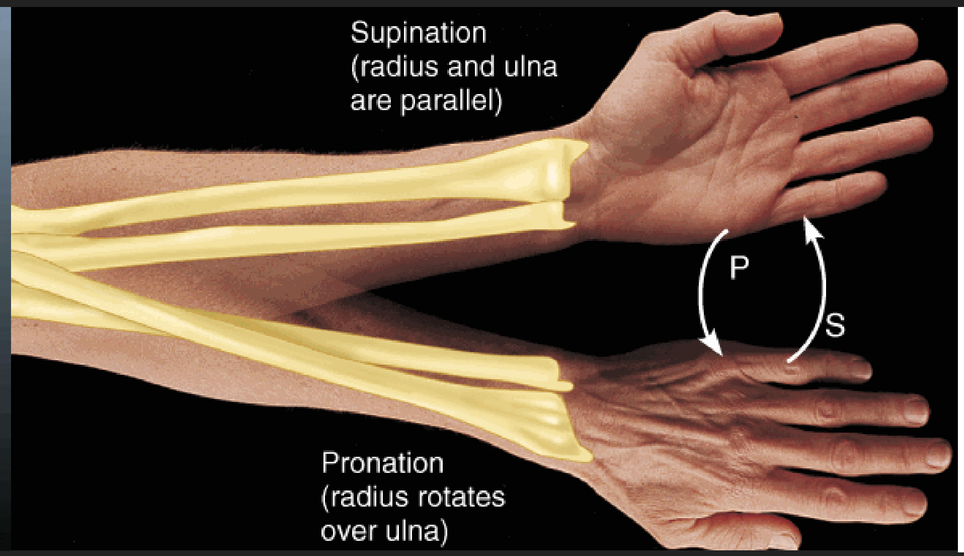
- Made up of three bones, the humerus, ulna, and radius
- Hinge joint serves to allow motion primarily in one plane

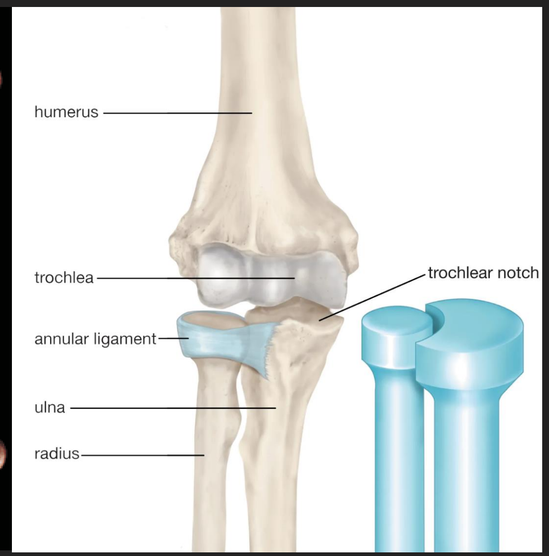




Forearm Twisting

- Known as pronation and supination of the forearm
- Setup Roll bones to achieve the twist distribution of arms
- Pivot joint allows only rotary movement around a single axis





Procedural Joints

- RBF joints: driven by RBF solver and get the transformation based on pose interpolation
- SDK joints: controlled by set-driven keys
- Constraint joints: runtime-aim/point/orient/parent constraints, etc.

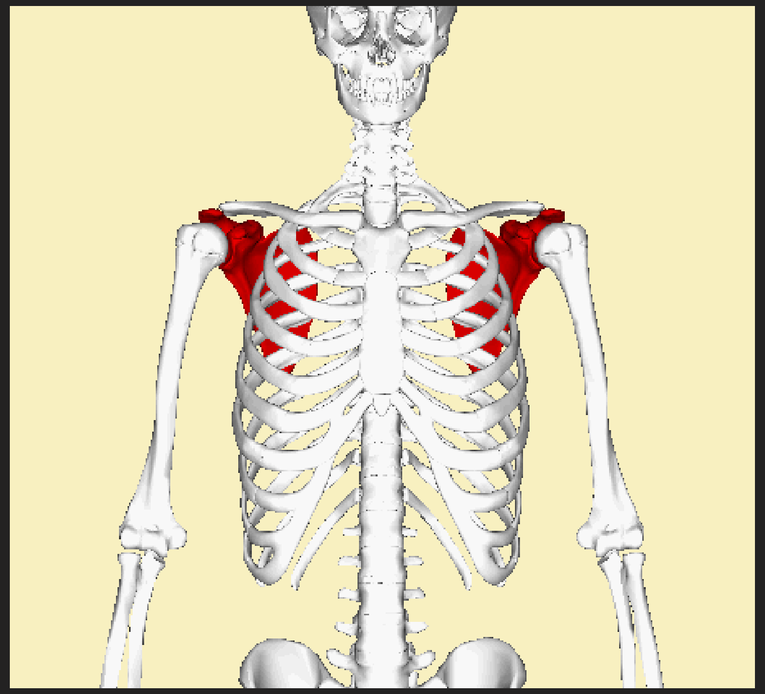
Thor's shoulder Pad is driven by RBF joints

Tyr's elbow is corrected by SDK joints

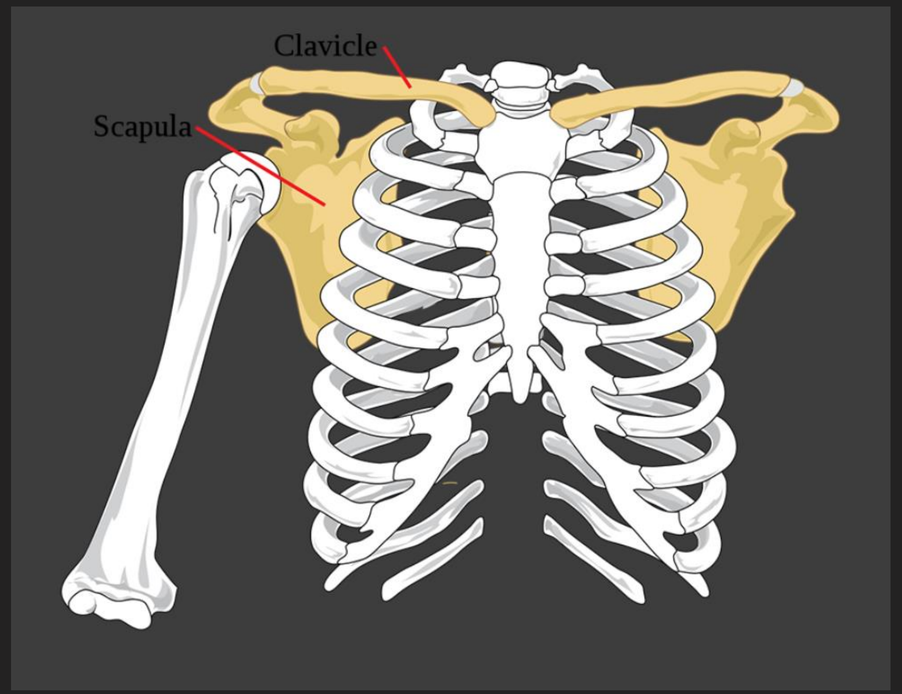
Advantages of Procedural Joints

- Save memory usage and the overall size of the game will be reduced
- Better support procedural animation
- Friendly to animators

Scapula Bone



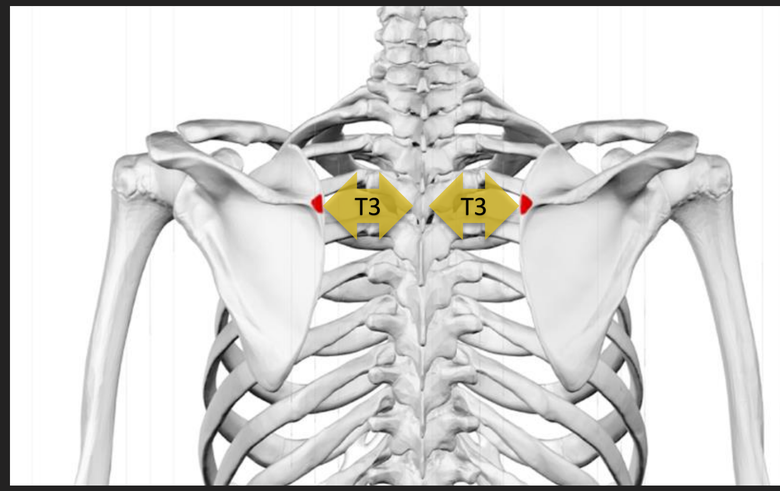
Position of the scapula shown in red



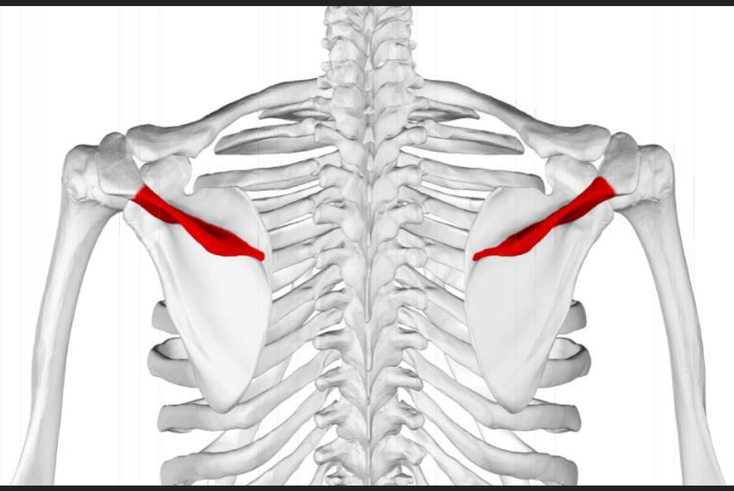
Anterior view of the scapula and clavicle

Spine of Scapula

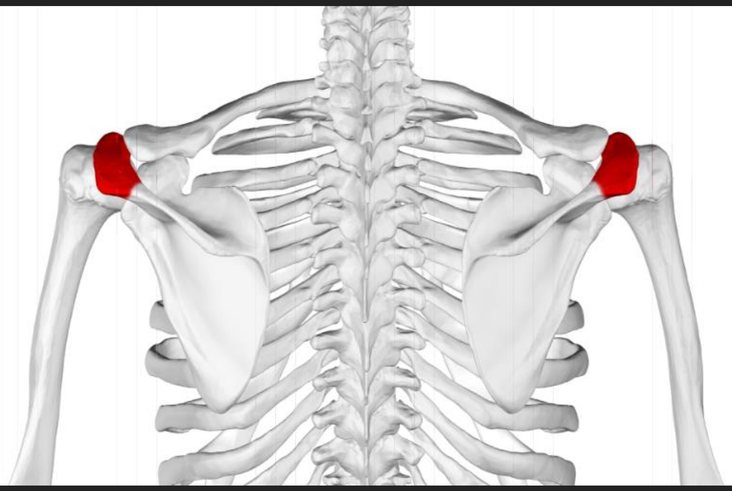
Starts at the root of the spine of the scapula and ends in the acromion



Root of spine shown in red



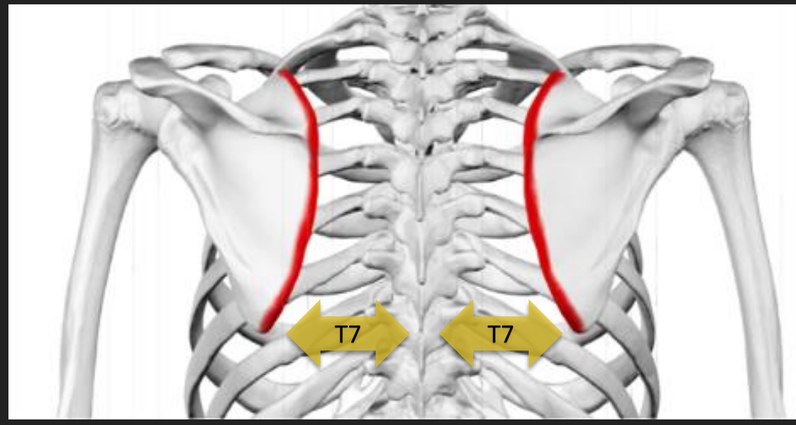
Spine of scapula shown in red



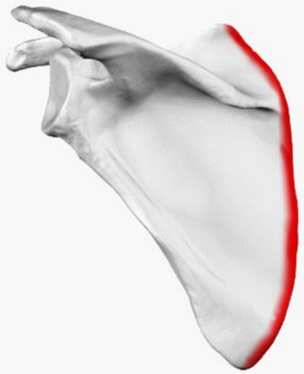
Acromion of each scapula shown in red

Medial border

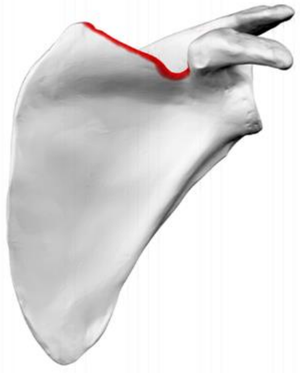
Starts from the superior angle to the inferior angle



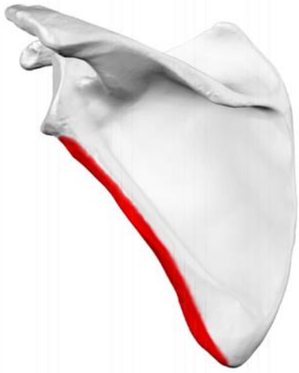
Medial border shown in red



Medial border



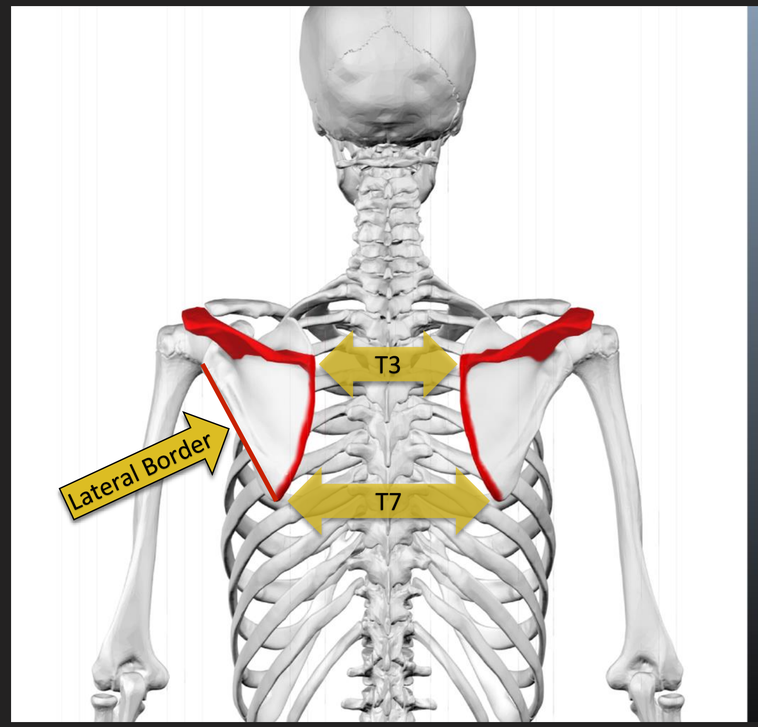
Superior border



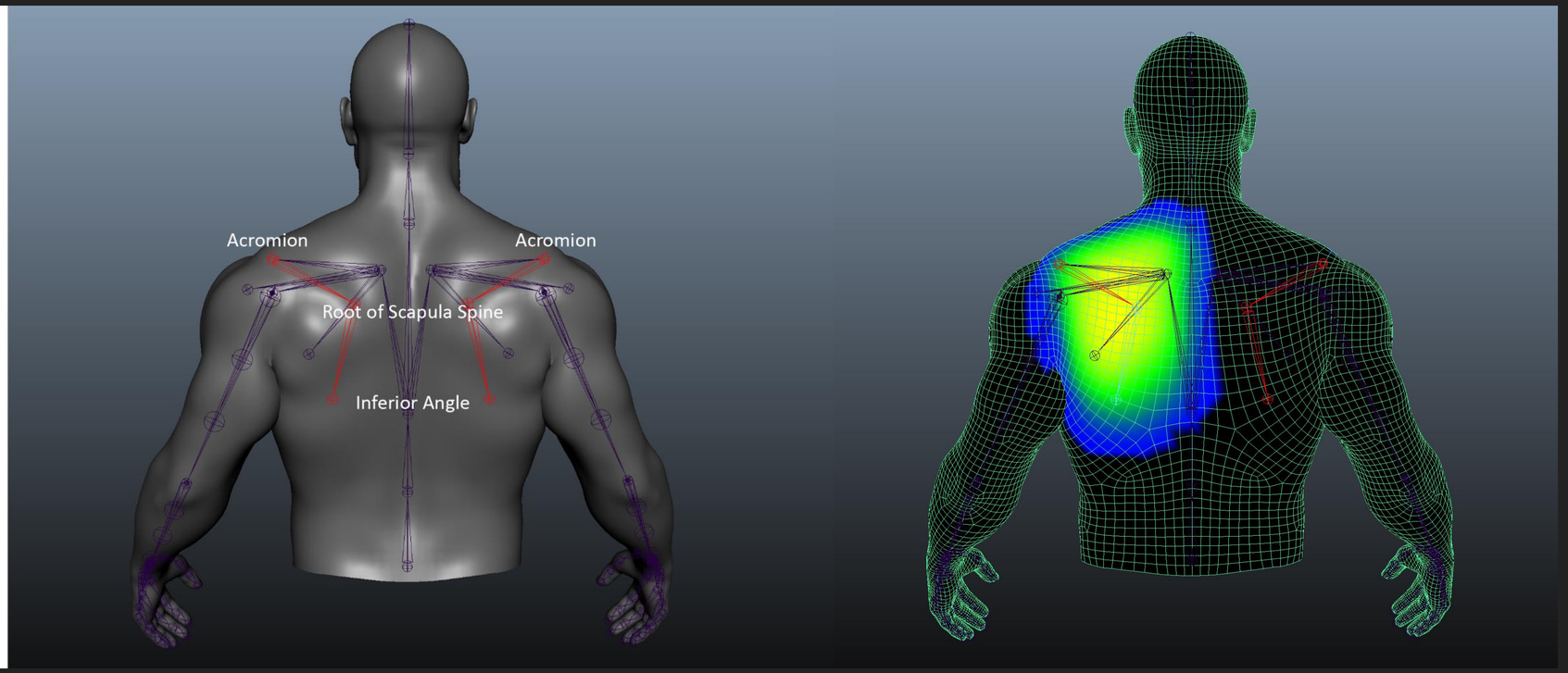
Axillary border

Inferior angle

Scapula Joint Positioning



Start at the acromion, extends along the spine of t he scapula and down to the inferior angle

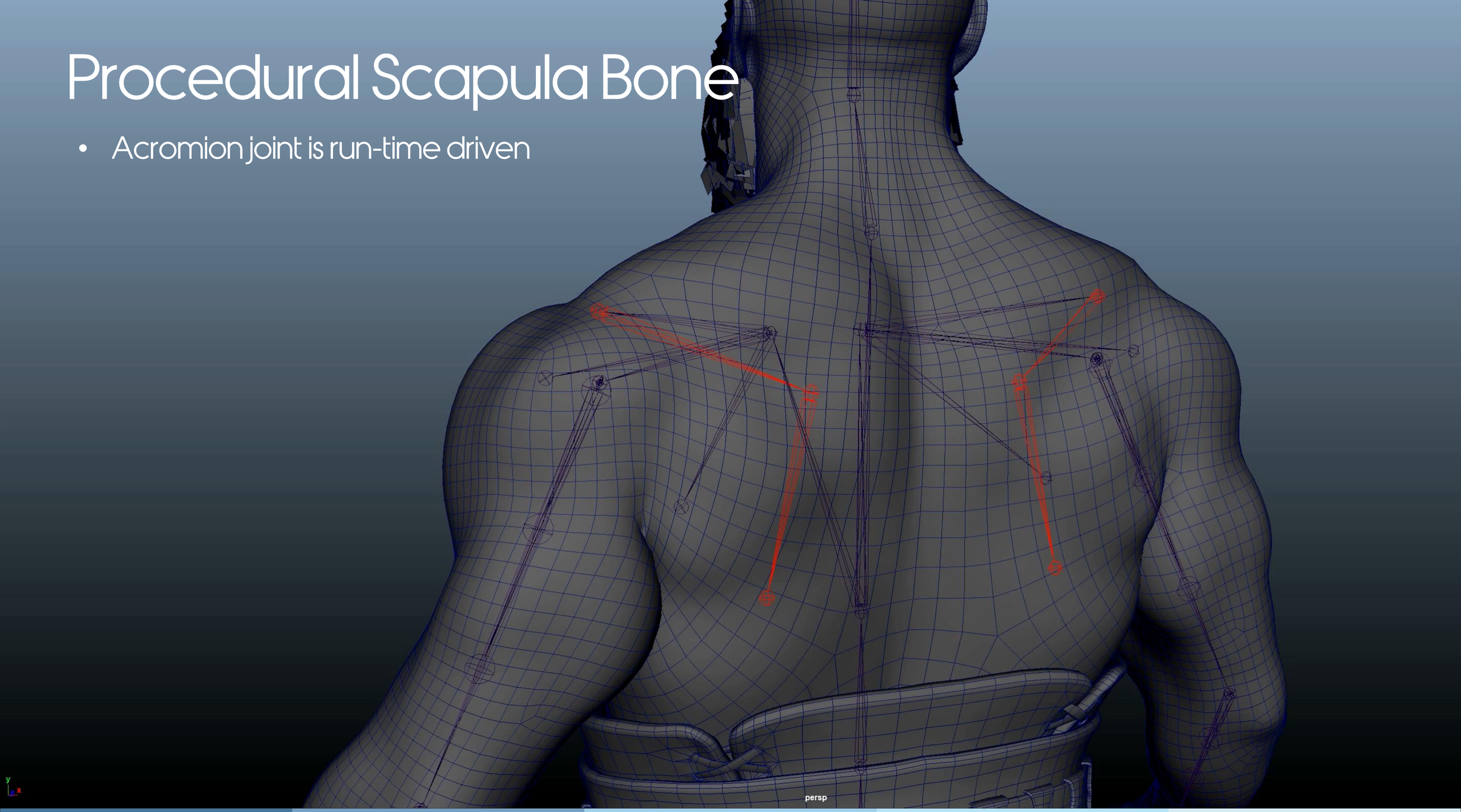


The three scapula joints are shown in red

Skin weights of the left scapula joint

Procedural Scapula Bone

- Acromion joint is run-time driven



Muscular System

- Focus on skeletal superficial muscles
- Learn the anatomy and function of the muscles

Superficial Back Muscles

- Trapezius
- Latissimus Dorsi
- Teres Major

Arm Muscles

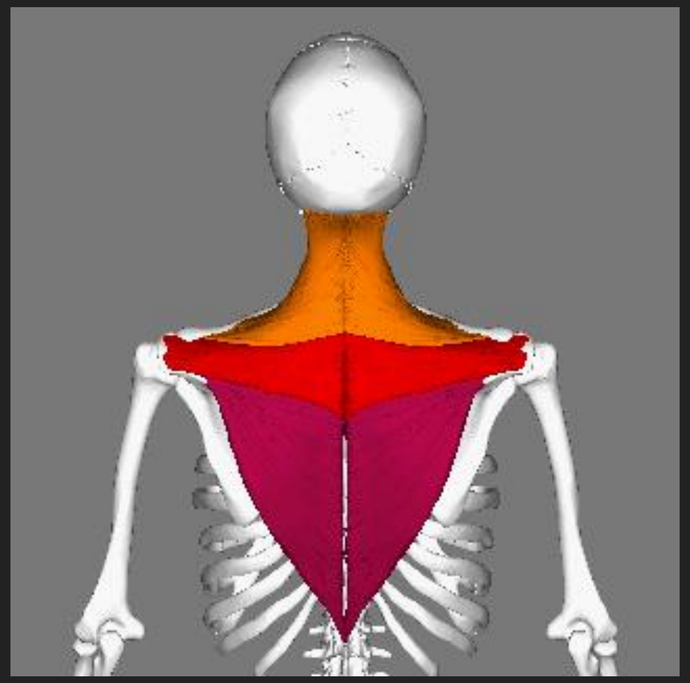
- Deltoid
- Triceps
- Biceps

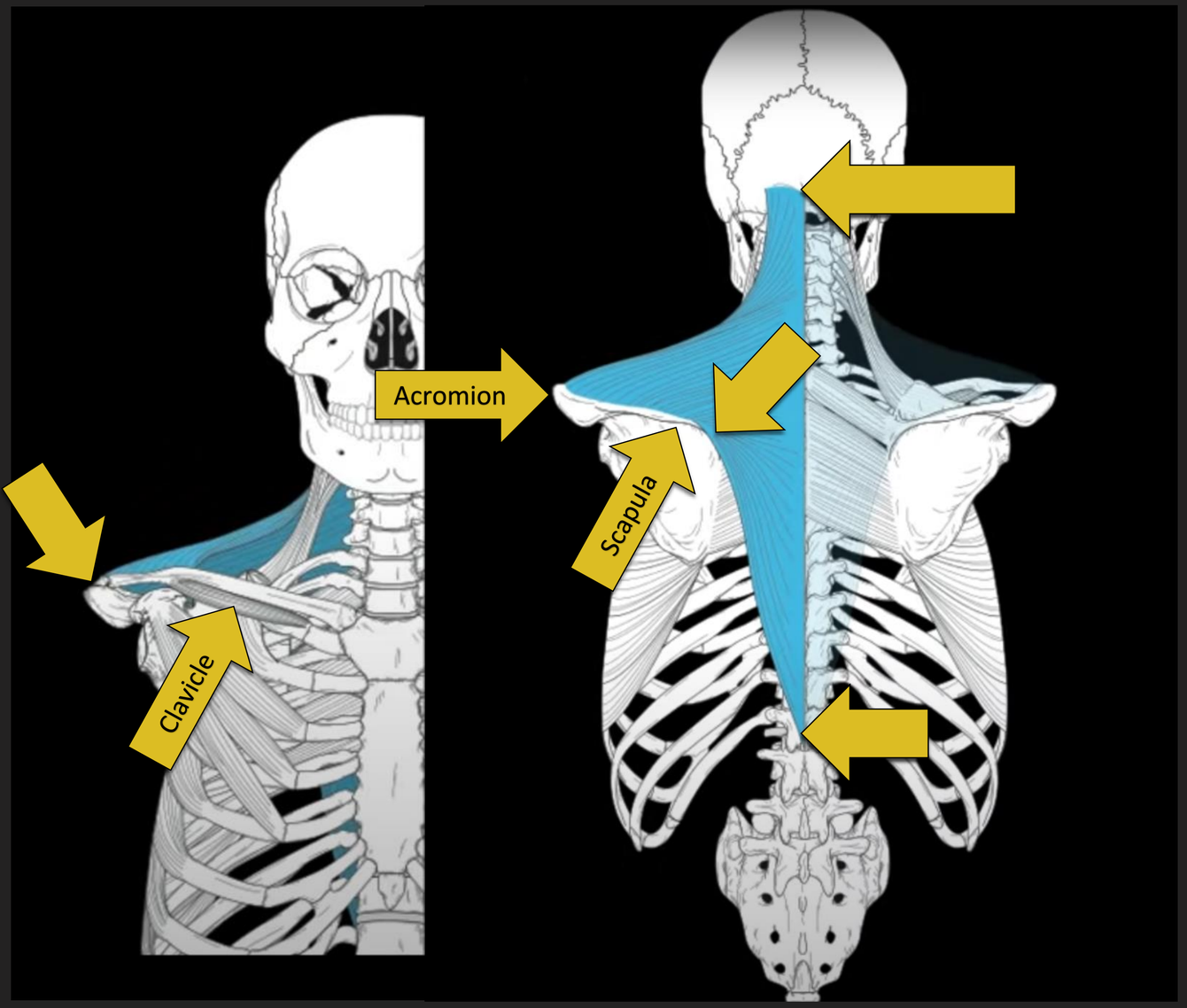
Chest Muscles

- Pectoralis Major

Trapezius Muscle

- Descending Part (superior fibers)
- Transverse Part (middle fibers)
- Ascending Part (inferior fibers)

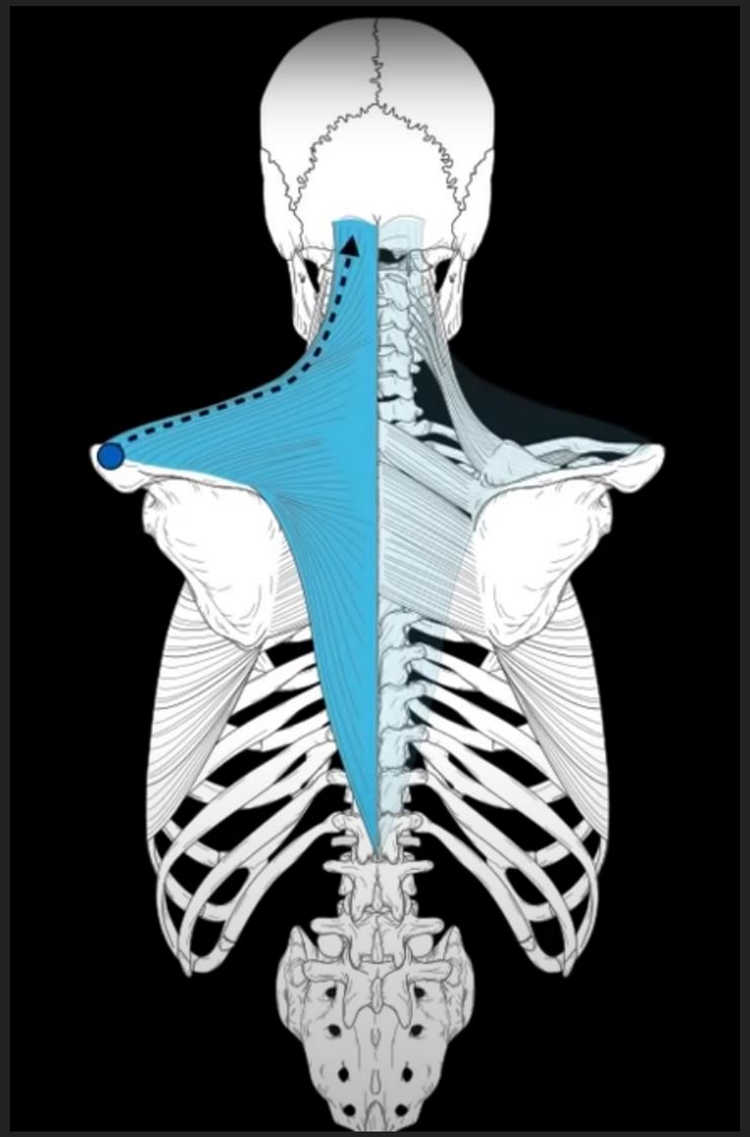


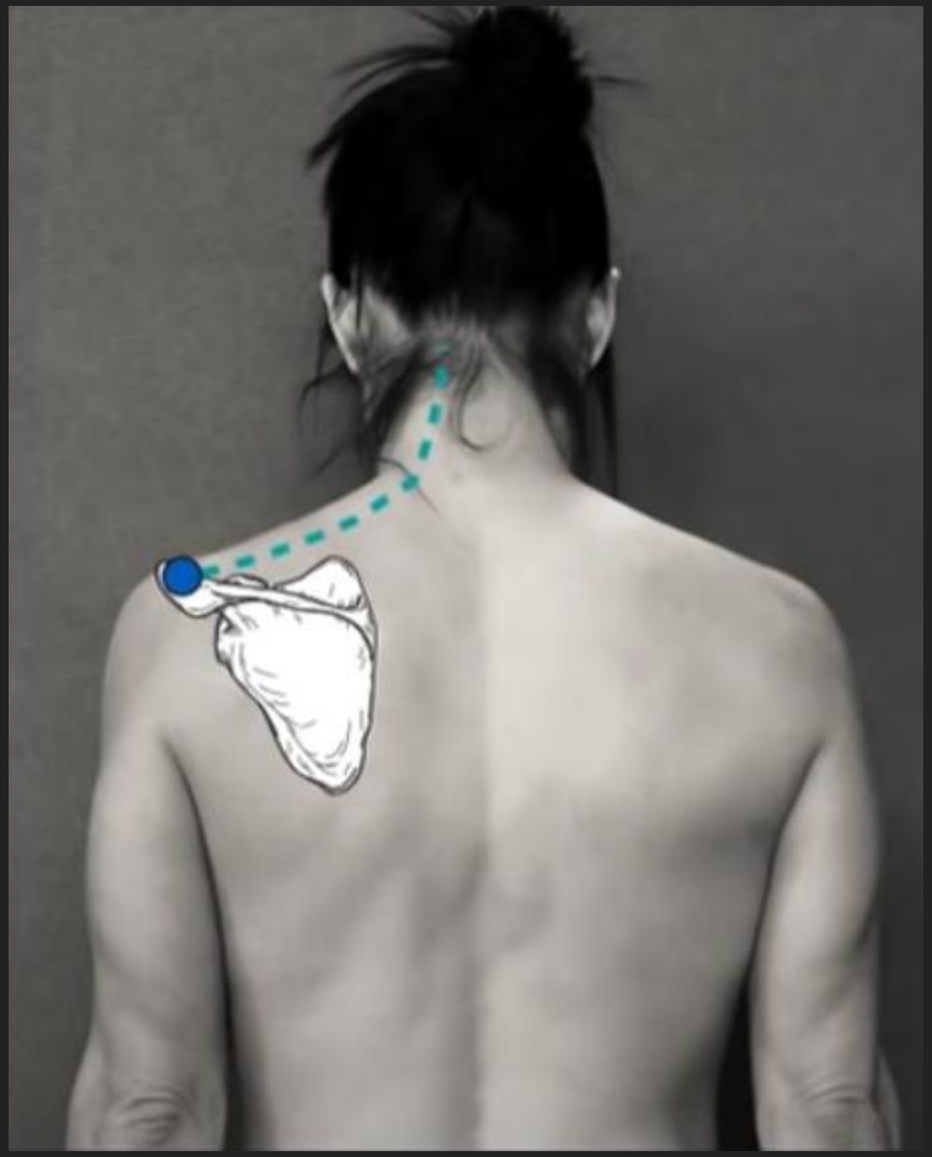


Descending Part

Origin: spinous process of the vertebrae C1 – C7

Insertion: lateral third of the clavicle

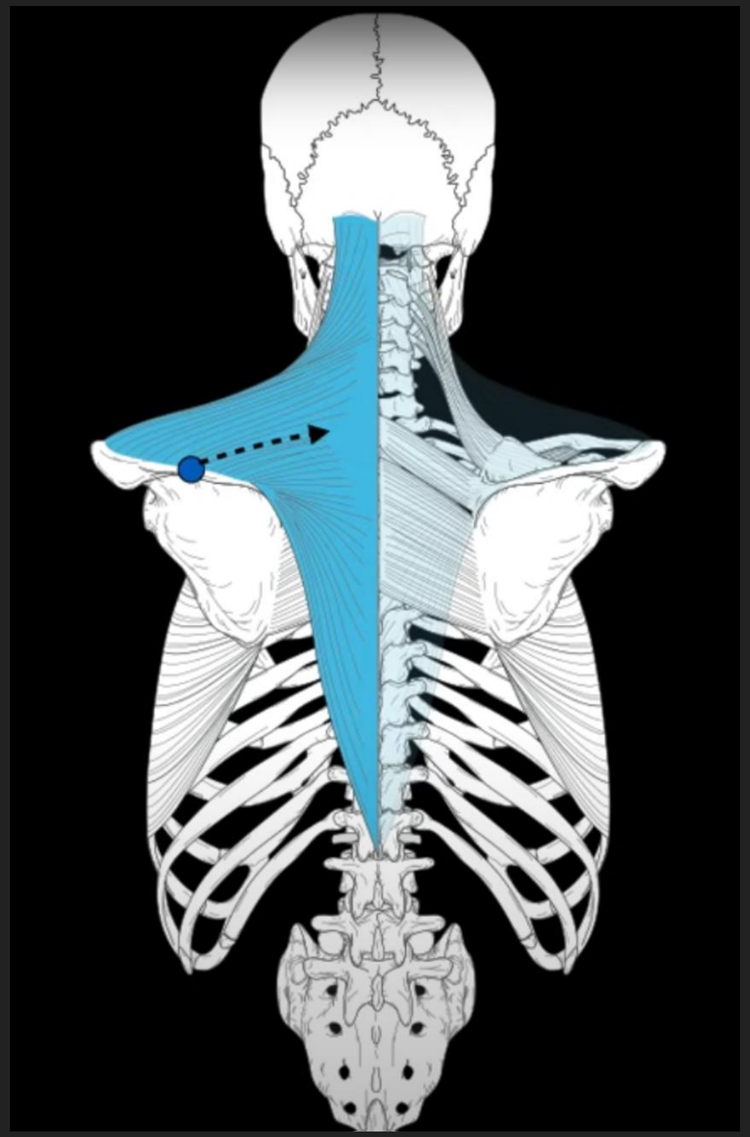


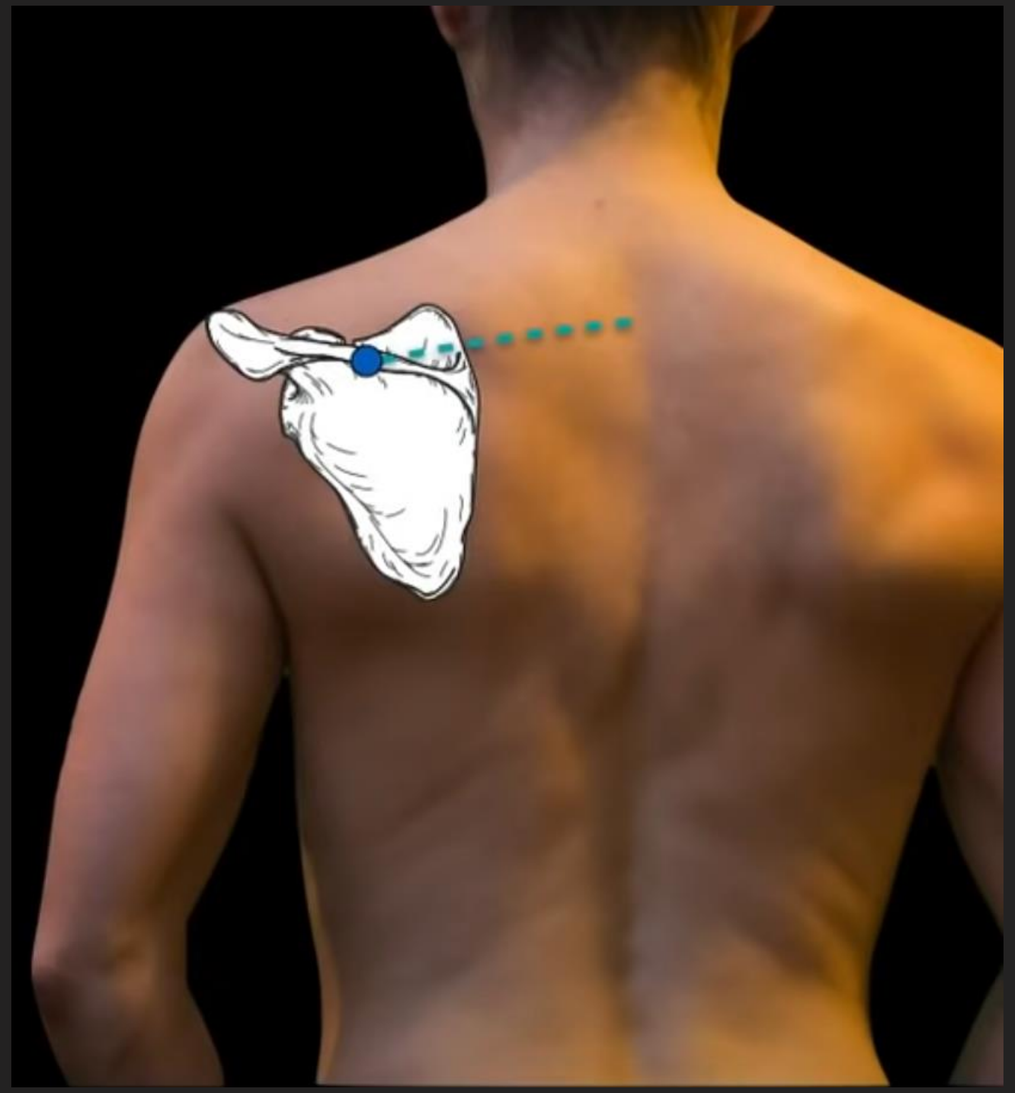


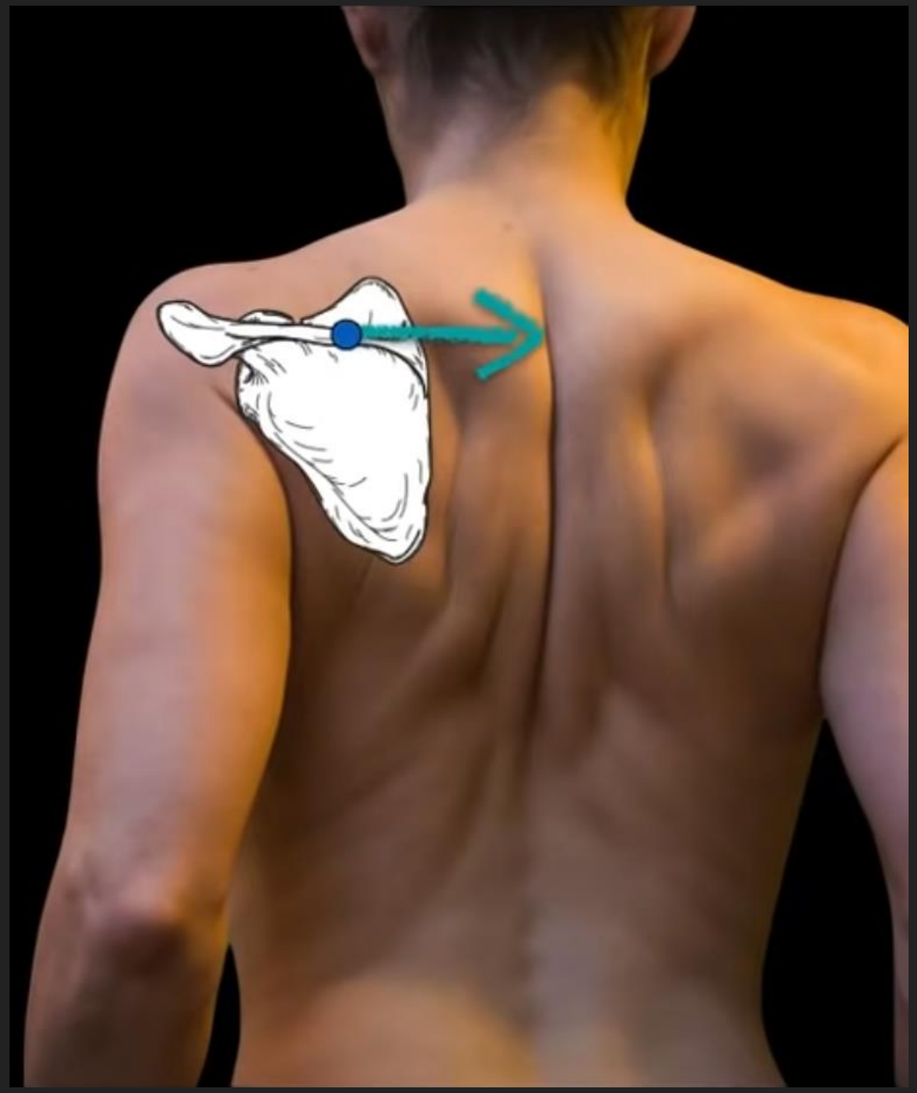
Traverse Part

Origin: spinous process of vertebrae T1 – T4 (C7 – T3)

Insertion: medial acromial margin



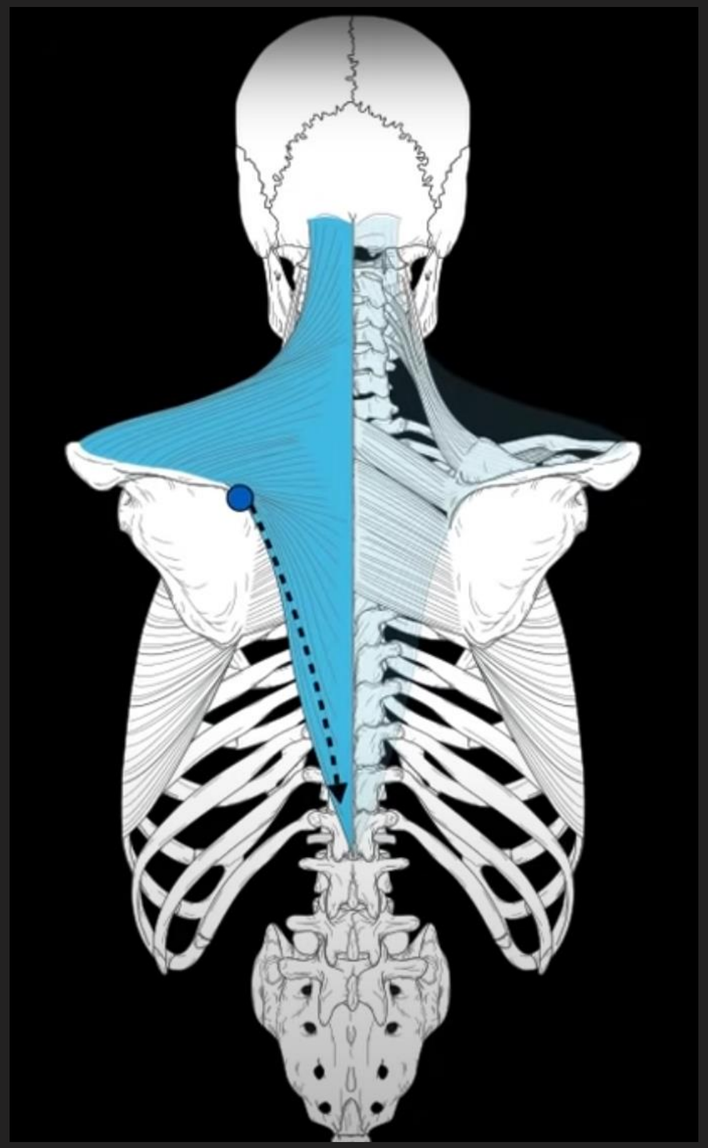


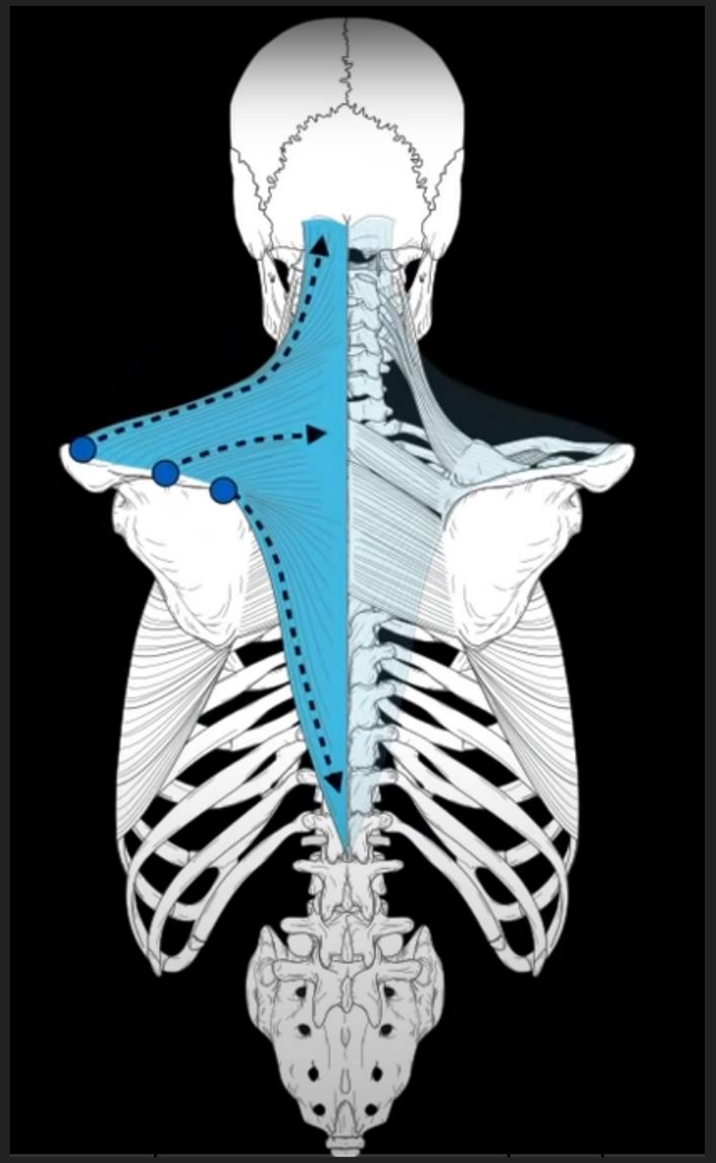


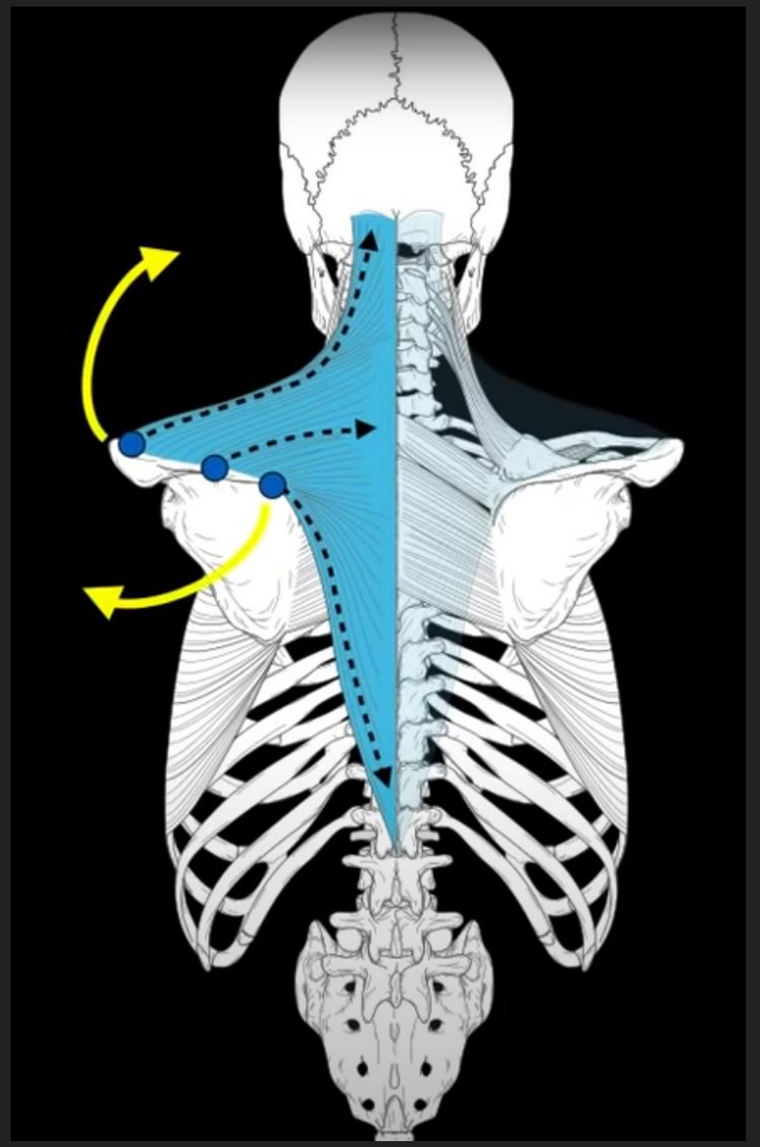
Ascending Part

Origin: spinous process of thoracic vertebrae (T4 – T12)

Insertion: medial end of the scapular spine







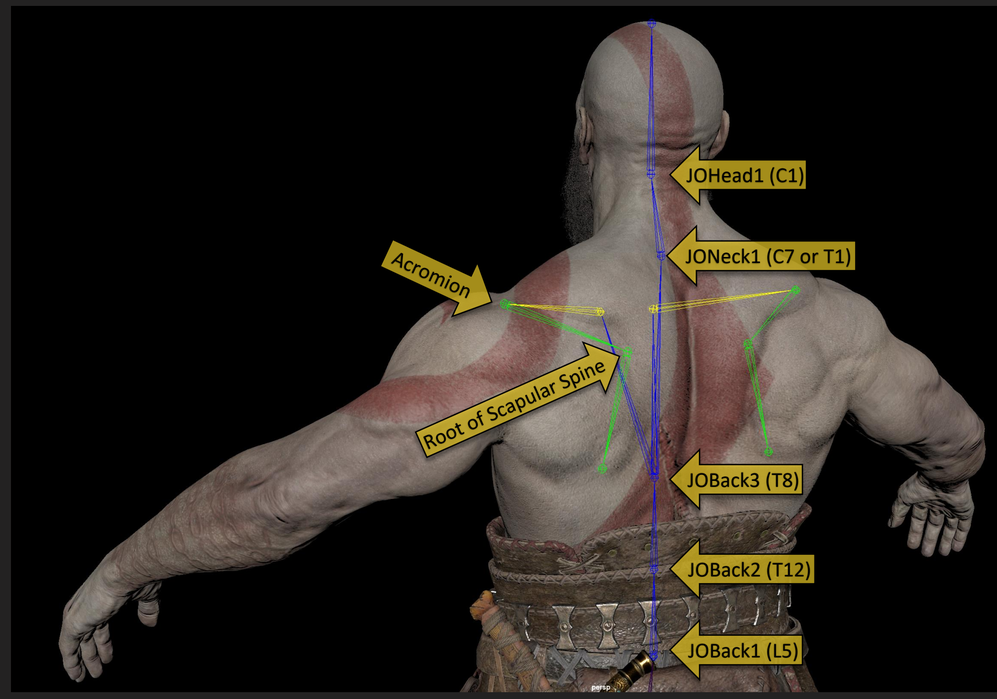
Bone Mapping

Origin:

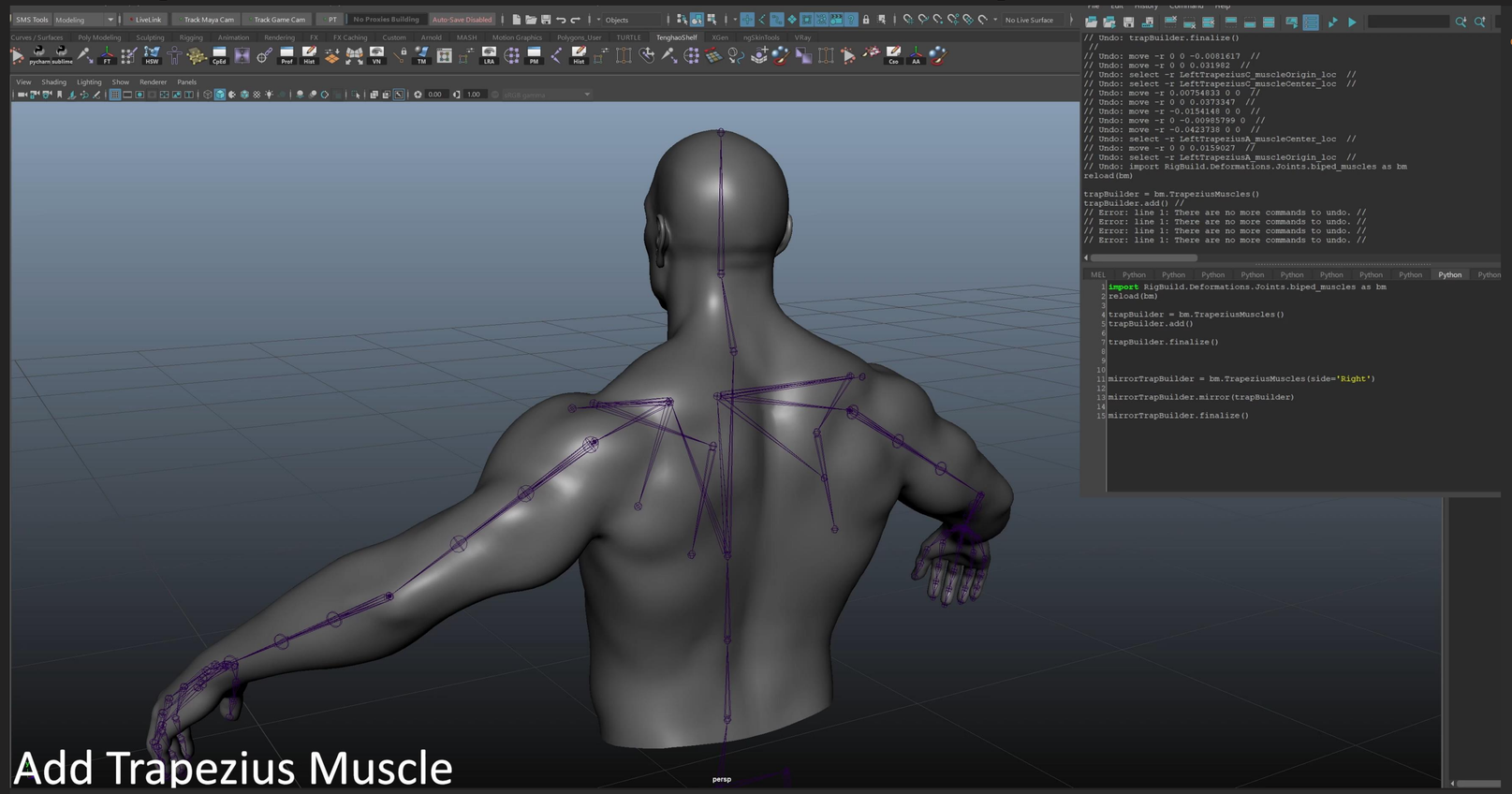
Descending Part: spinous process C1 – C7
Traverse Part: spinous process T1 – T4
Ascending Part: spinous process T4 – T12

Insertion:

Descending Part: lateral third of the clavicle
Traverse Part: medial acromial margin
Ascending Part: medial end of the scapular spine



Trapezius Muscle Component

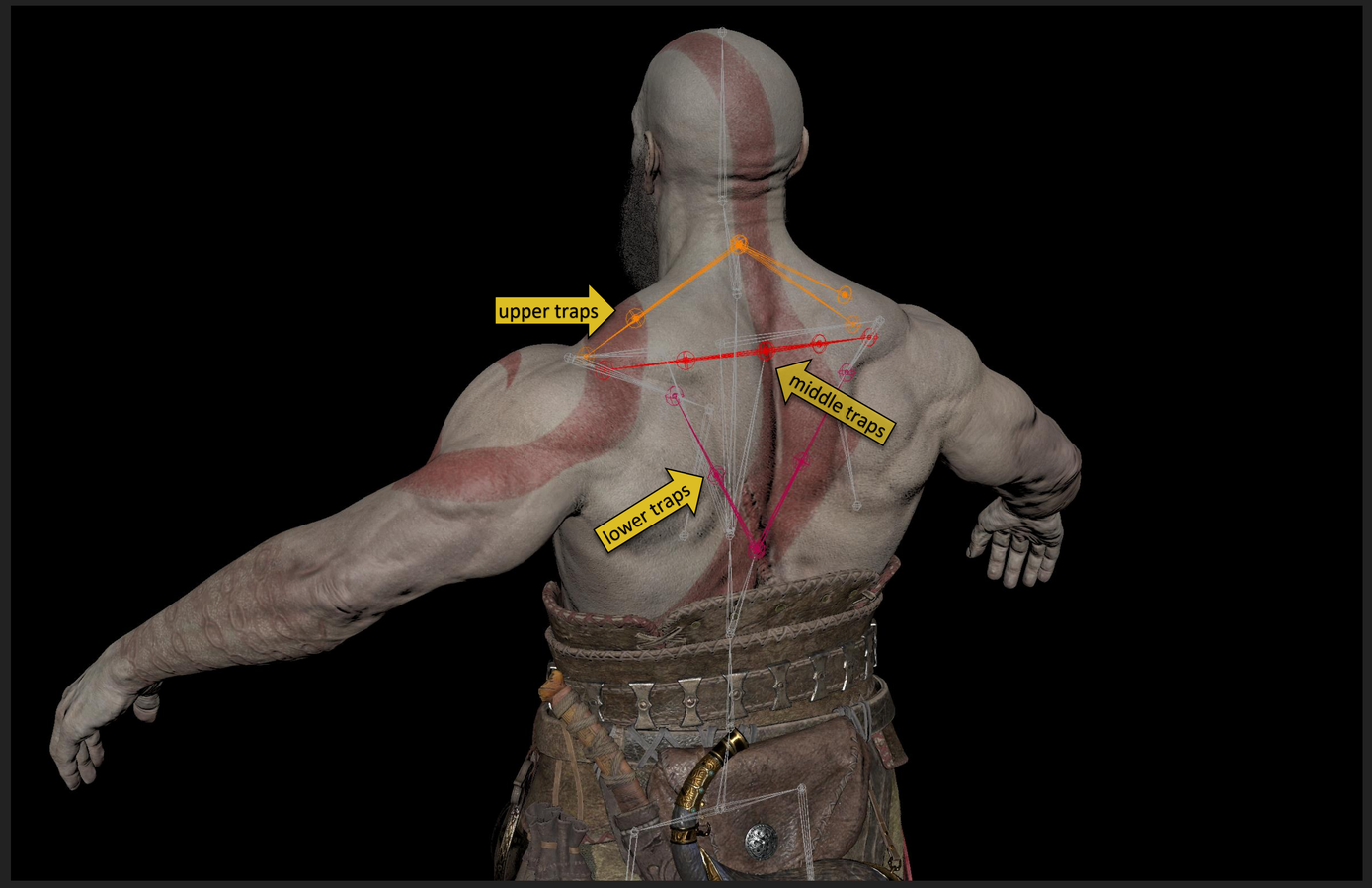


Add Trapezius Muscle

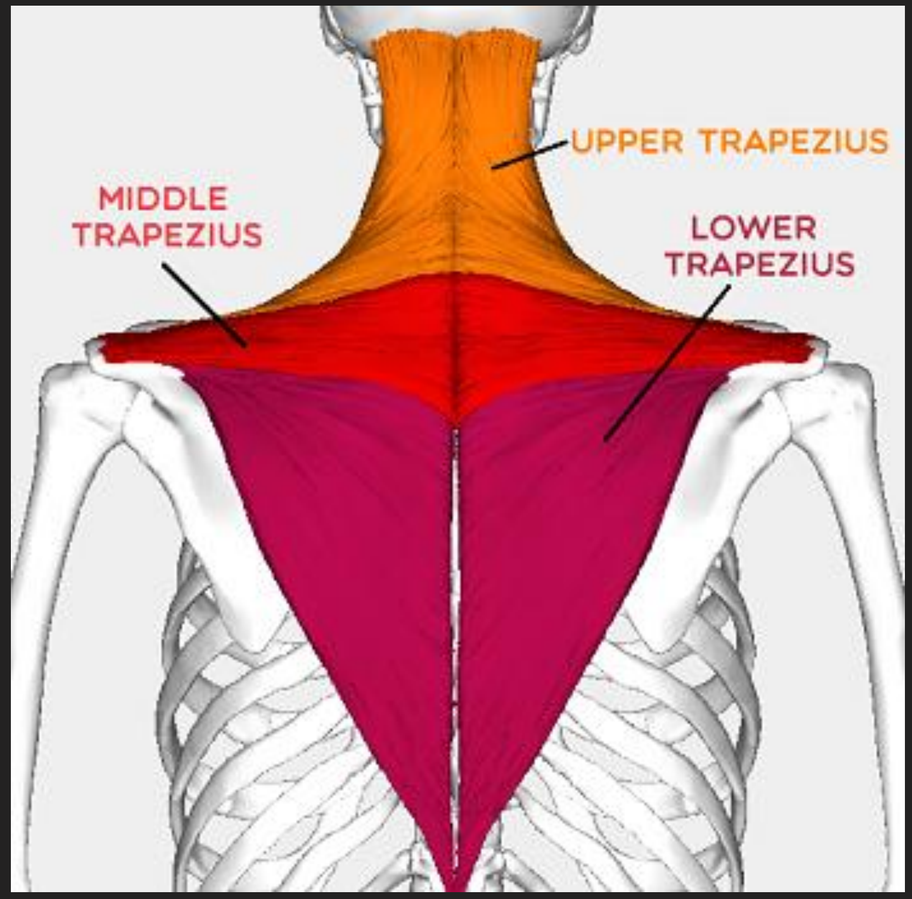
```
# create muscle joint groups for left side
trapBuilder = TrapeziusMuscles(side='Left')
trapBuilder.add()
trapBuilder.finalize()


# mirror the left traps to the right side
mirrorTrapBuilder = TrapeziusMuscles(side='Right')
mirrorTrapBuilder.mirror(trapBuilder)
mirrorTrapBuilder.finalize()
```

```
class TrapeziusMuscles(BipedMuscles):


    def add(self):
        # Descending Part: superior fibers of the trapezius
        # Origin: spinous process of C7 to the occipital bone (neck joint to head joint)
        neckJointPos = om.MVector(mc.xform(self.neckJoint, translation=True, q=True, ws=True))
        headJointPos = om.MVector(mc.xform(self.headJoint, translation=True, q=True, ws=True))
        averagePos = (neckJointPos + headJointPos) / 2.0
        neckJointWorldMatrix = mc.getAttr('{0}.worldMatrix'.format(self.neckJoint))
        offsetVector = om.MVector(neckJointWorldMatrix[8: 11]) * 0.02
        trapeziusAOrigin = offsetVector + averagePos

        # Insertion: lateral third of the clavical
        clavicalJointPos = om.MVector(mc.xform(self.clavicalJoint, translation=True, q=True, ws=True))
        acromoinJointPos = om.MVector(mc.xform(self.acromionJoint, translation=True, q=True, ws=True))
        offsetVector = (clavicalJointPos - acromoinJointPos) / 6.0
        trapeziusAInsertion = acromoinJointPos + offsetVector

        self.trapeziusA = mb.MuscleJoint.createFromAttachObjs(muscleName='{0}{1}A'.format(self.side, self.name),
                                                              originAttachObj=self.neckJoint,
                                                              insertionAttachObj=self.clavicalJoint,
                                                              compressionFactor=0.5,
                                                              stretchFactor=1.5,
                                                              bulgeVector=[0.0, 0.0])
        self.muscleParts.append(self.trapeziusA)
        mc.xform(self.trapeziusA.originLoc, translation=trapeziusAOrigin, worldSpace=True)
        mc.xform(self.trapeziusA.insertionLoc, translation=trapeziusAInsertion, worldSpace=True)
        # Transverse Part: middle fibers of the trapezius
        scapulaJointPos = om.MVector(mc.xform(self.scapulaJoint, translation=True, q=True, ws=True))
        # Origin: Broad aponeurosis at spinous processes of vertebrae T1-T4 (or C7-T3)
        averagePos = (neckJointPos + scapulaJointPos) / 2.0
        trapeziusBOrigin = om.MVector(neckJointPos.x, averagePos.y, averagePos.z)
        # Insertion: Medial aspect of acromion, Superior crest of spine of scapula
        offsetVector = (scapulaJointPos - acromoinJointPos) / 4.0
        trapeziusBInsertion = offsetVector + acromoinJointPos
        self.trapeziusB = mb.MuscleJoint.createFromAttachObjs(muscleName='{0}{1}B'.format(self.side, self.name),
                                                              originAttachObj=self.spine3Joint,
                                                              insertionAttachObj=self.acromionJoint,
                                                              compressionFactor=0.5,
                                                              stretchFactor=1.5,
                                                              bulgeVector=[0.0, 0.0])
        self.muscleParts.append(self.trapeziusB)
        mc.xform(self.trapeziusB.originLoc, translation=trapeziusBOrigin, worldSpace=True)
        mc.xform(self.trapeziusB.insertionLoc, translation=trapeziusBInsertion, worldSpace=True)
        # Ascending Part: inferior fibers of the trapezius
        # Origin: Arise from the spinous processes of the remaining thoracic vertebrae (T4-T12)
        # roughly is T8 at the same level as the xiphisternum: (JOBack3)
        spine3JointPos = om.MVector(mc.xform(self.spine3Joint, translation=True, q=True, ws=True))
        trapeziusCOrigin = om.MVector(spine3JointPos.x, spine3JointPos.y, trapeziusBOrigin.z)
        # Insertion: Medial end of spine of scapula
        trapeziusCInsertion = offsetVector * 3 + acromoinJointPos
        self.trapeziusC = mb.MuscleJoint.createFromAttachObjs(muscleName='{0}{1}C'.format(self.side, self.name),
                                                              originAttachObj=self.spine2Joint,
                                                              insertionAttachObj=self.acromionJoint,
                                                              compressionFactor=0.5,
                                                              stretchFactor=1.5,
                                                              bulgeVector=[0.0, 0.0])
        self.muscleParts.append(self.trapeziusC)
        mc.xform(self.trapeziusC.originLoc, translation=trapeziusCOrigin, worldSpace=True)
        mc.xform(self.trapeziusC.insertionLoc, translation=trapeziusCInsertion, worldSpace=True)
```

Trapezius Muscle Component



Joint layout of trapezius muscle component

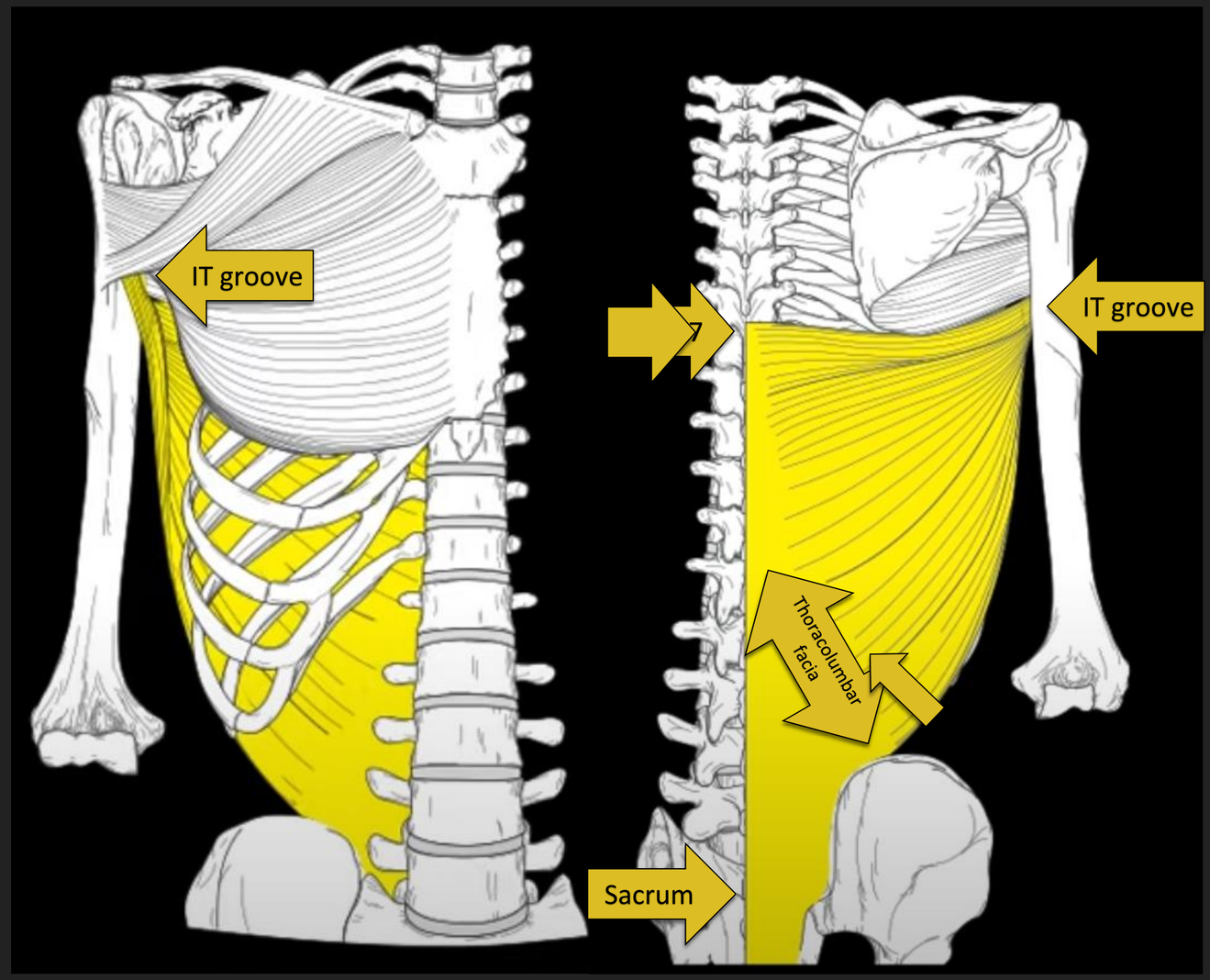


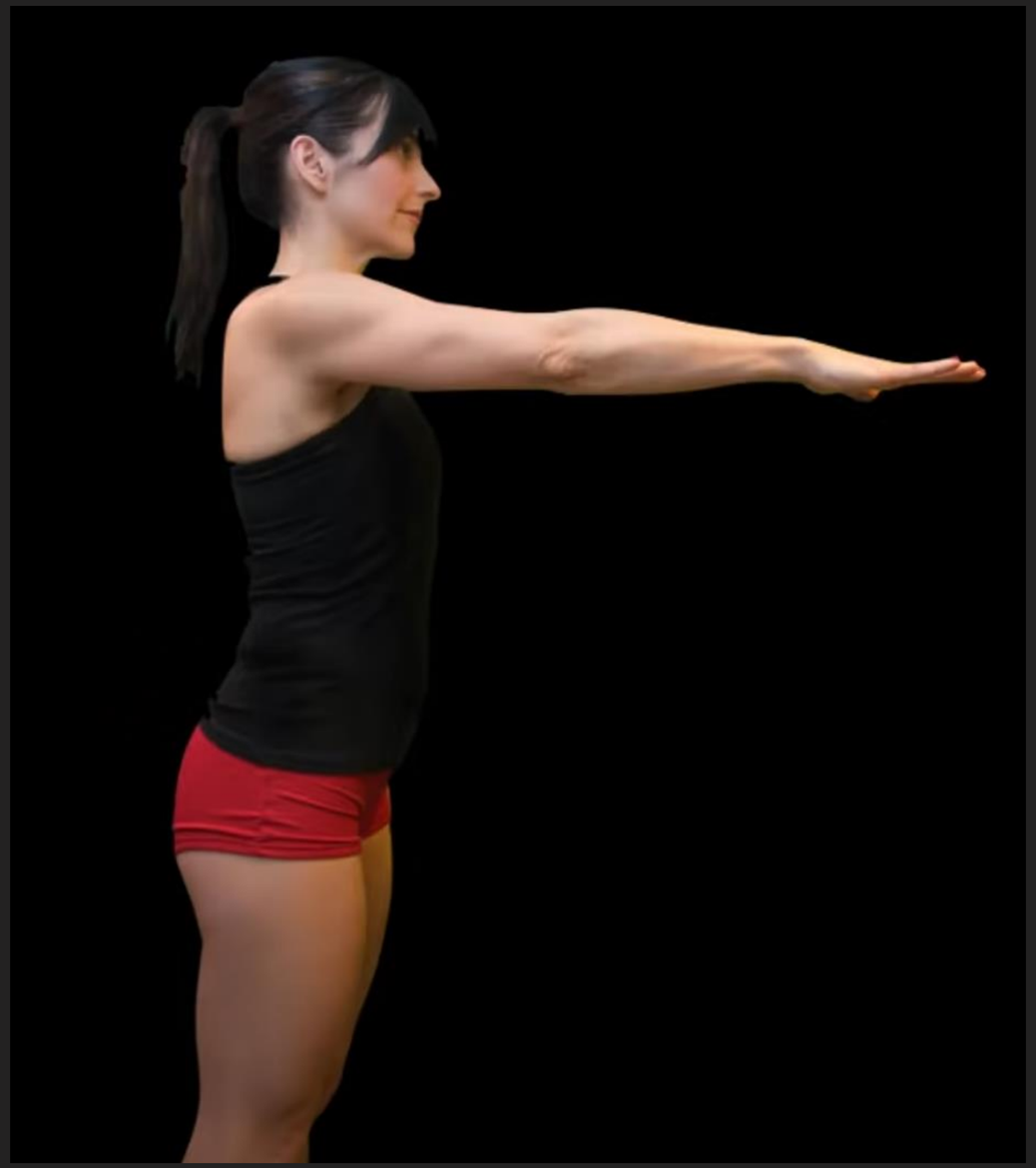
Trapezius muscle anatomy

Latissimus Dorsi Muscle

Origin: spinous process T7 – T12
thoracolumbar fascia

Insertion: intertubercular groove

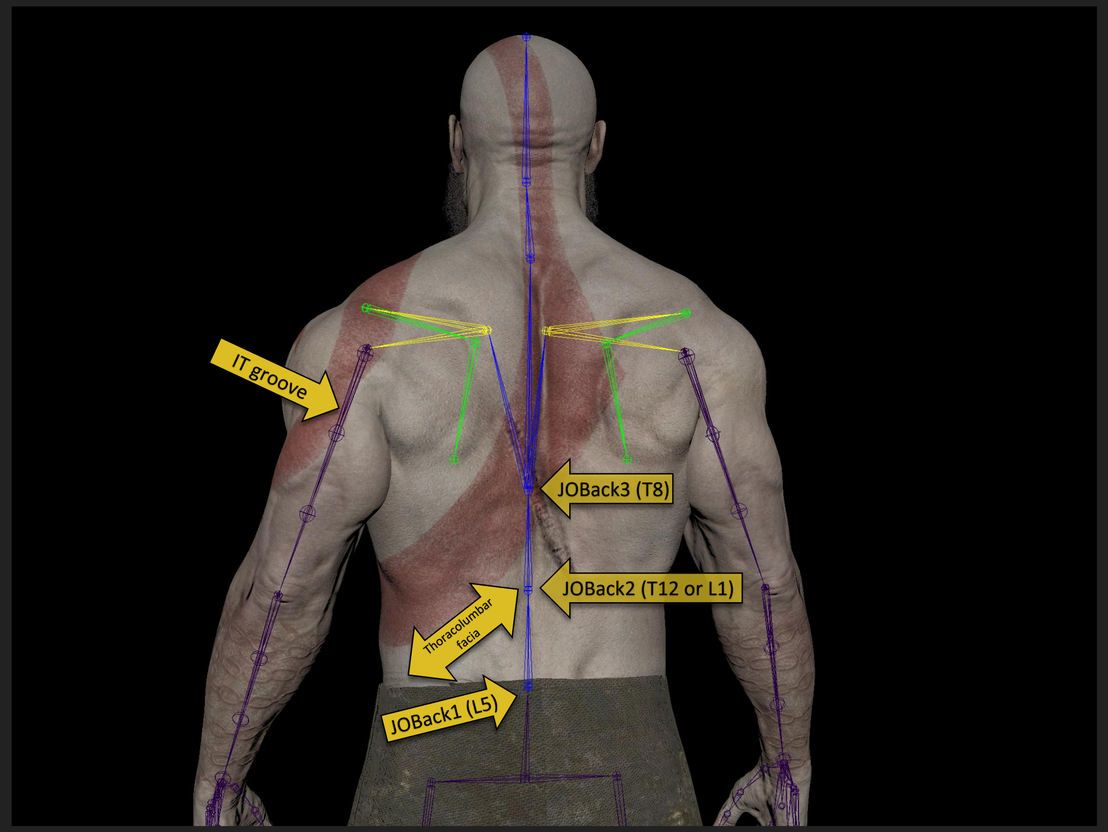




Bone Mapping

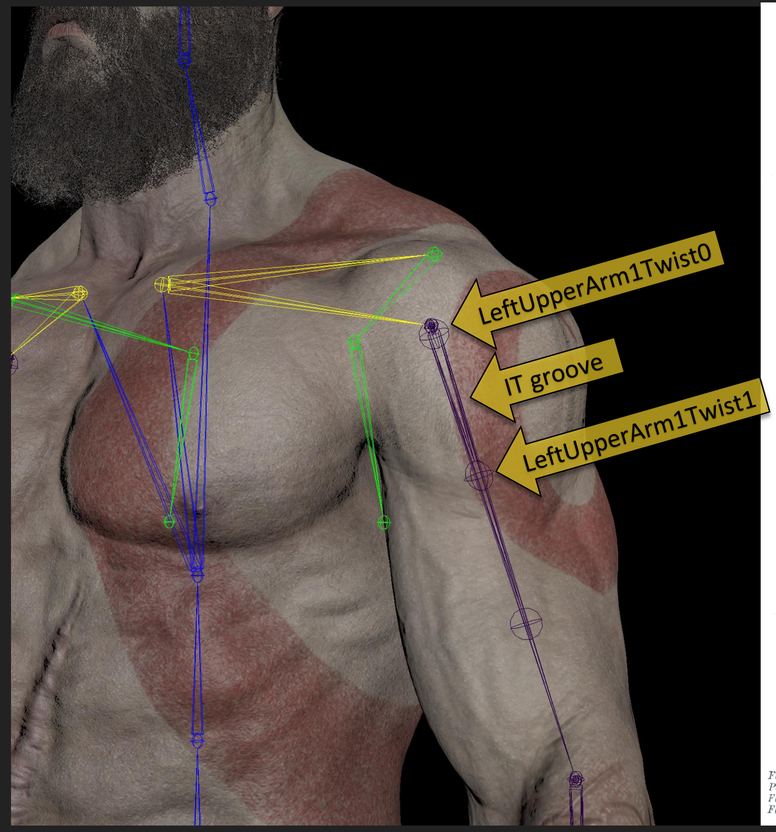
Origin:

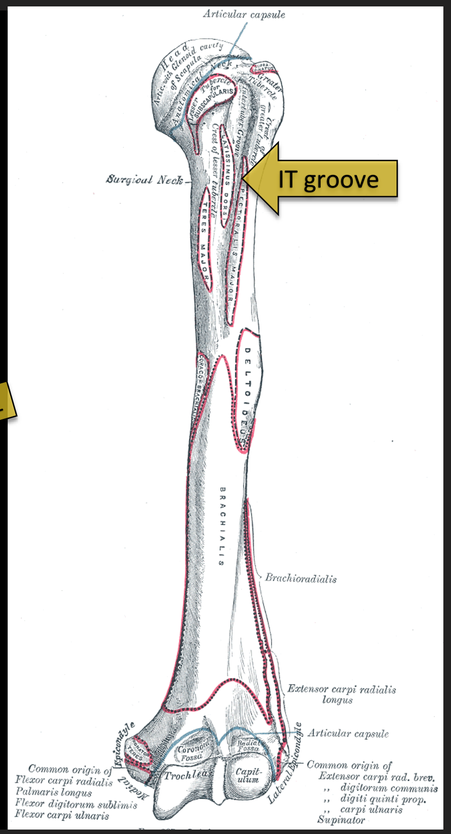
Vertebral Part: spinous process T7 — T12
Thoracolumbar fascia: spinous process (L1- L5)



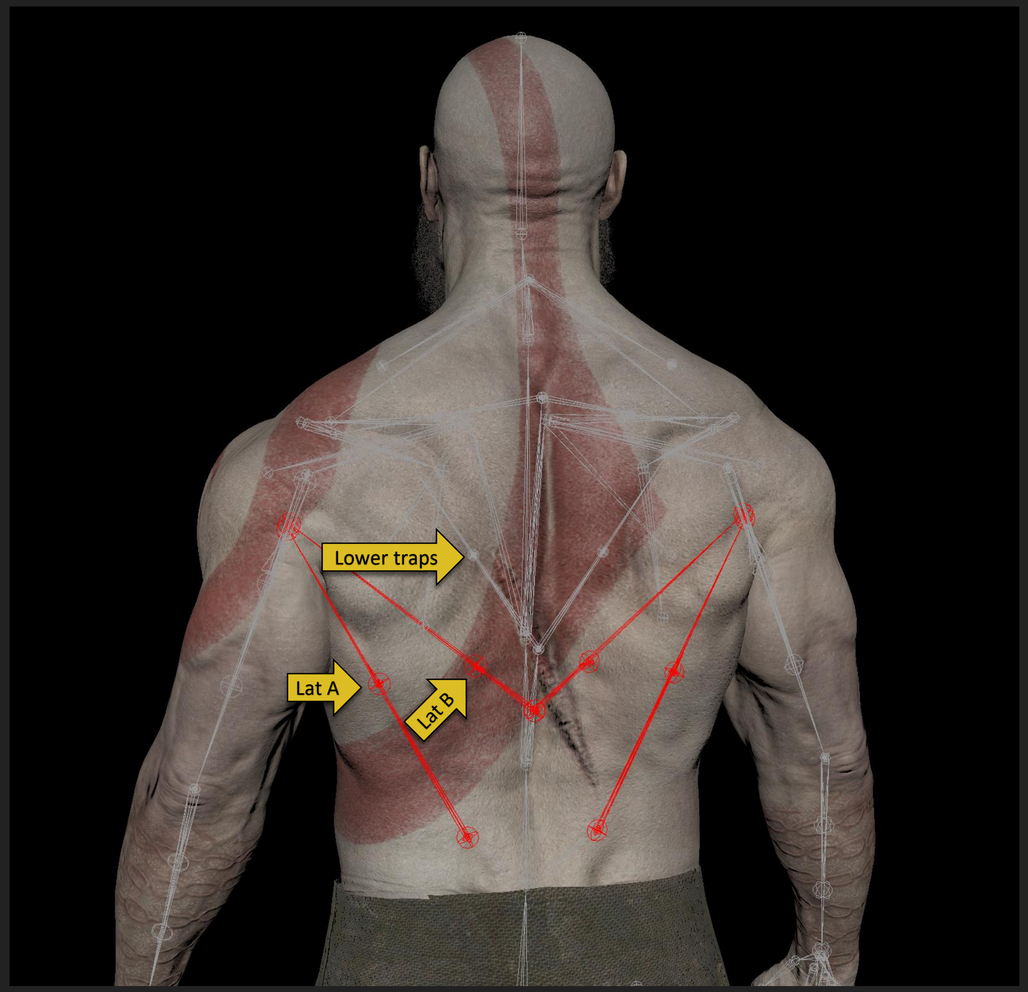
Insertion:

Intertubercular groove

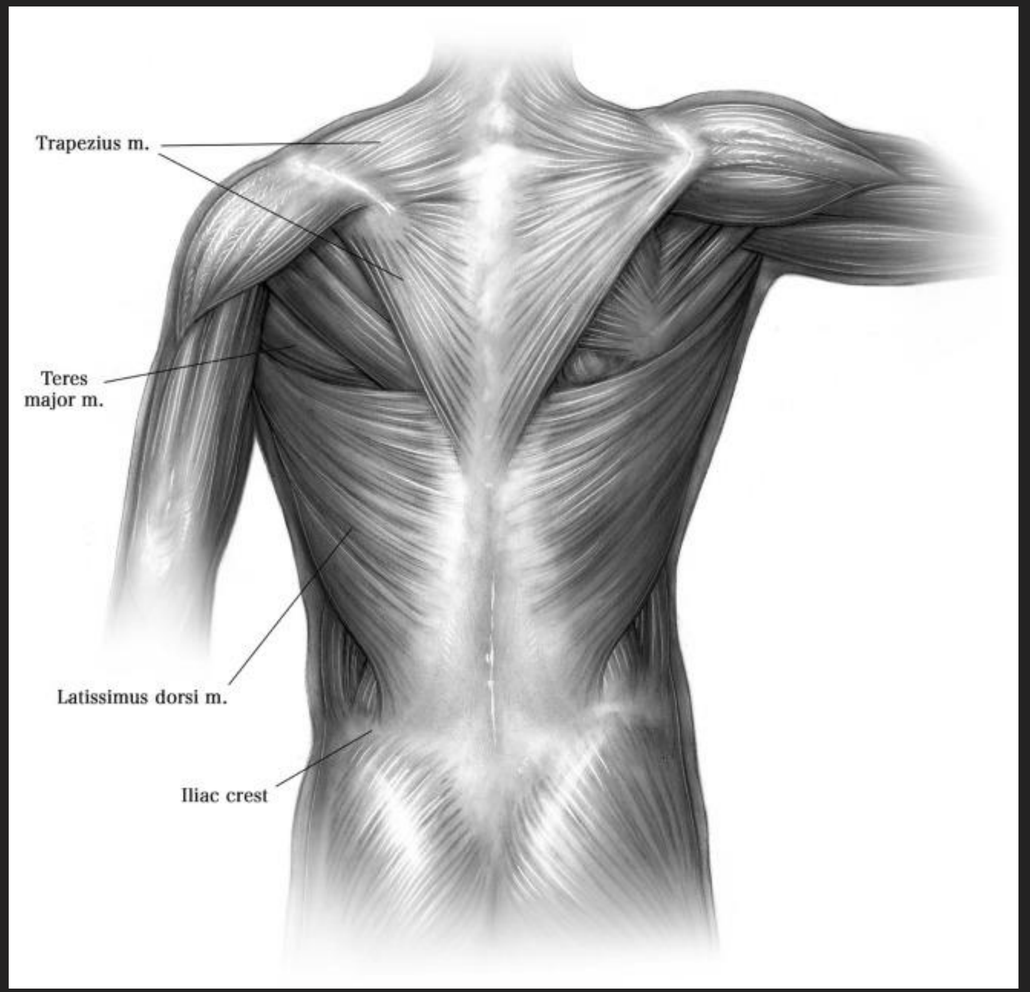




Latissimus Dorsi Muscle Component



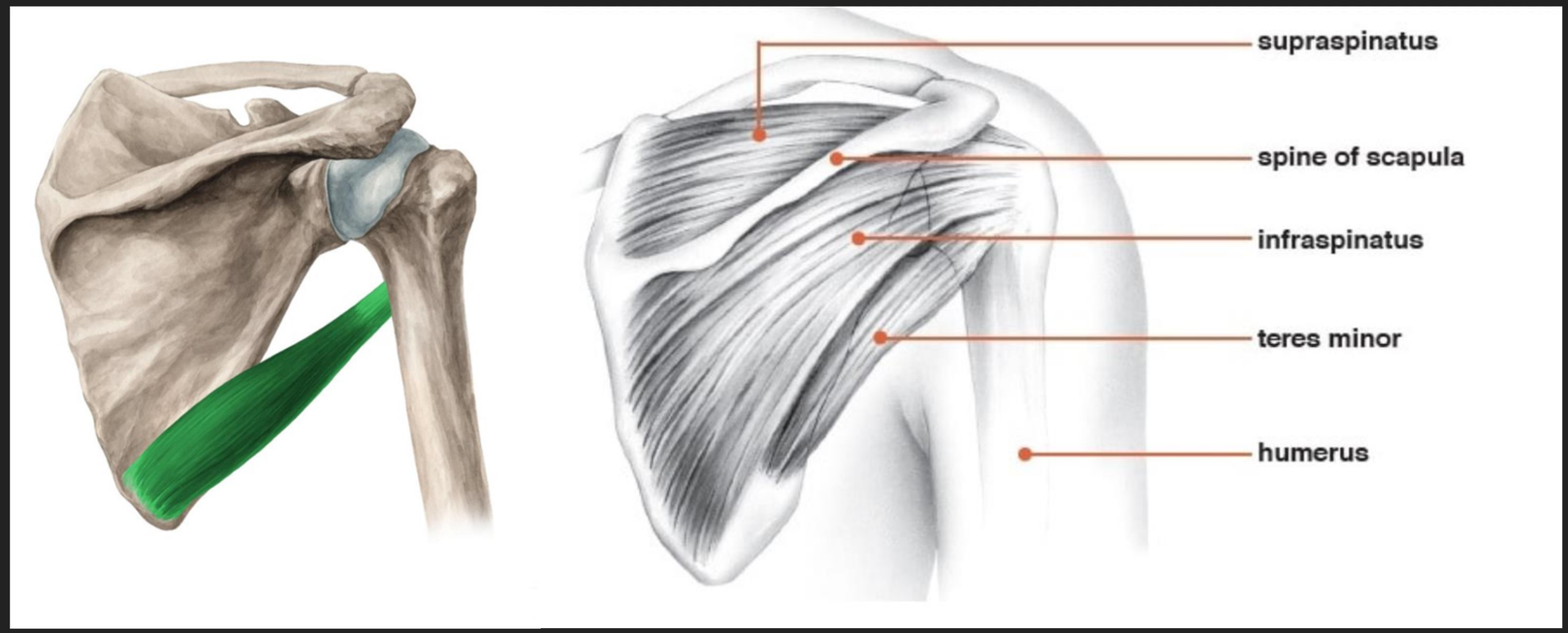
Joint layout of latissimus dorsi muscle component



Superficial back muscles anatomy

Scapula Muscles

- Teres major
- Supraspinatus
- Teres minor
- Infraspinatus



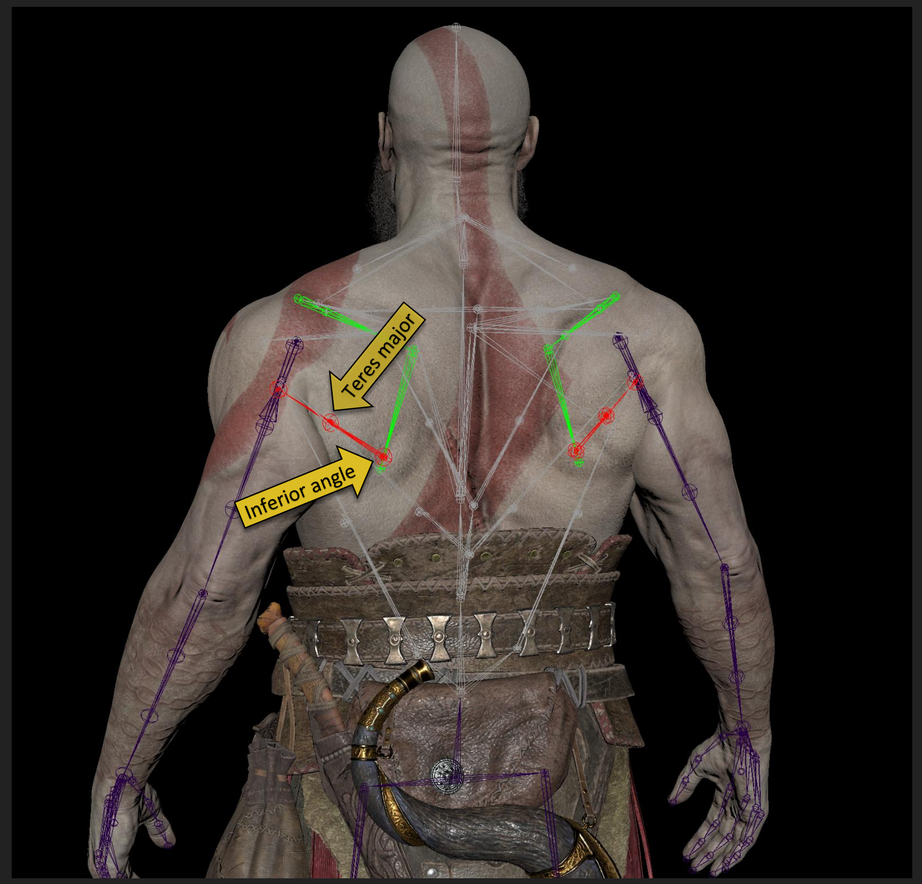
Teres major muscle in green

Rotate cuff muscles

Bone Mapping

Origin:

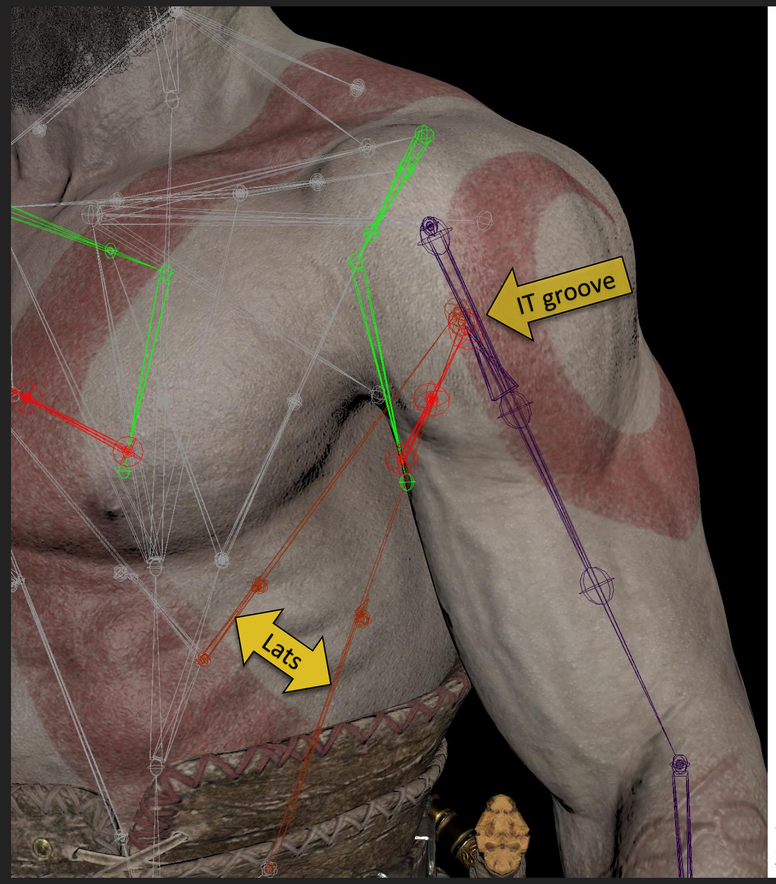
Inferior angle

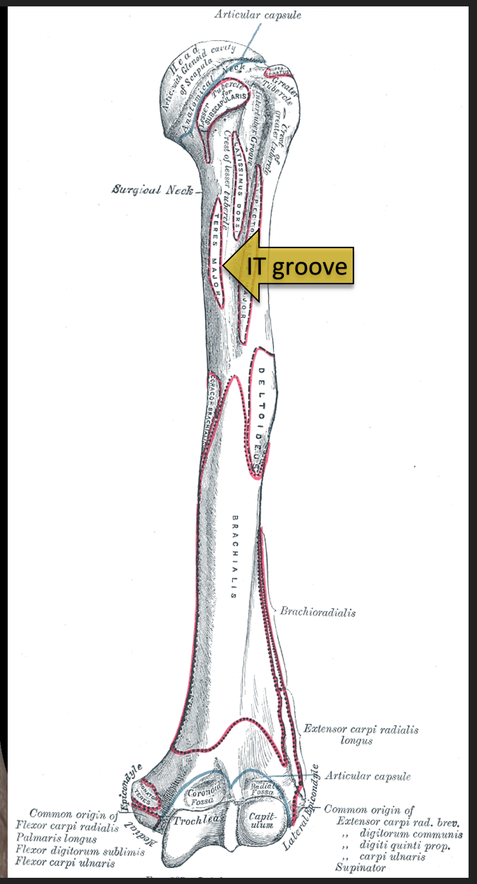


Joint layout of teres major muscle component

Insertion:

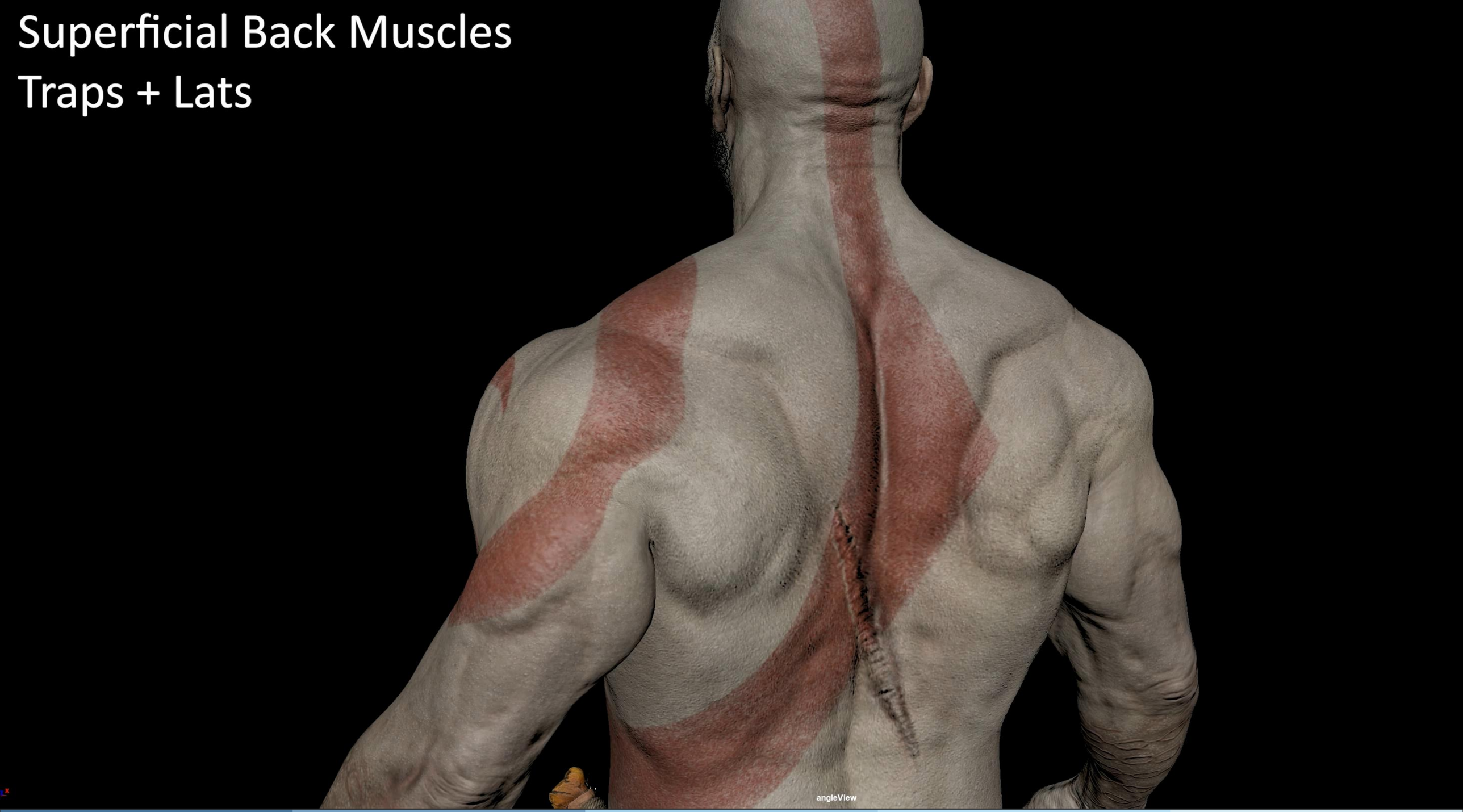
Intertubercular groove



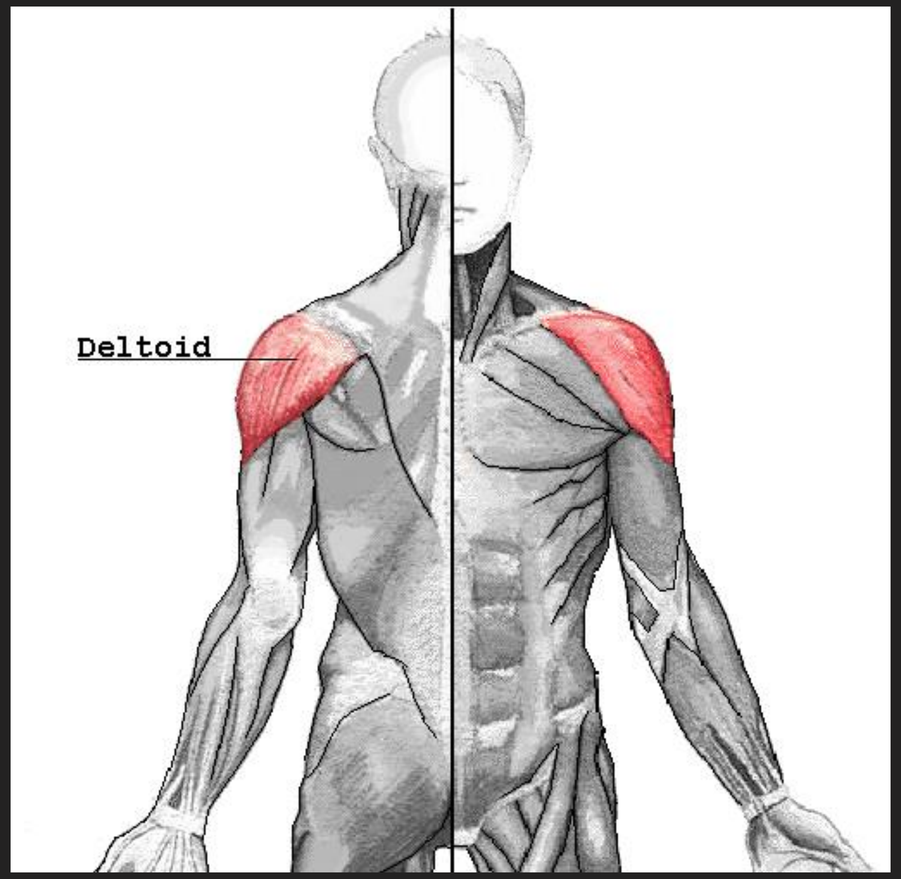


Common insertion for lats and teres major muscles

Superficial Back Muscles
Traps + Lats



Deltoid Muscle

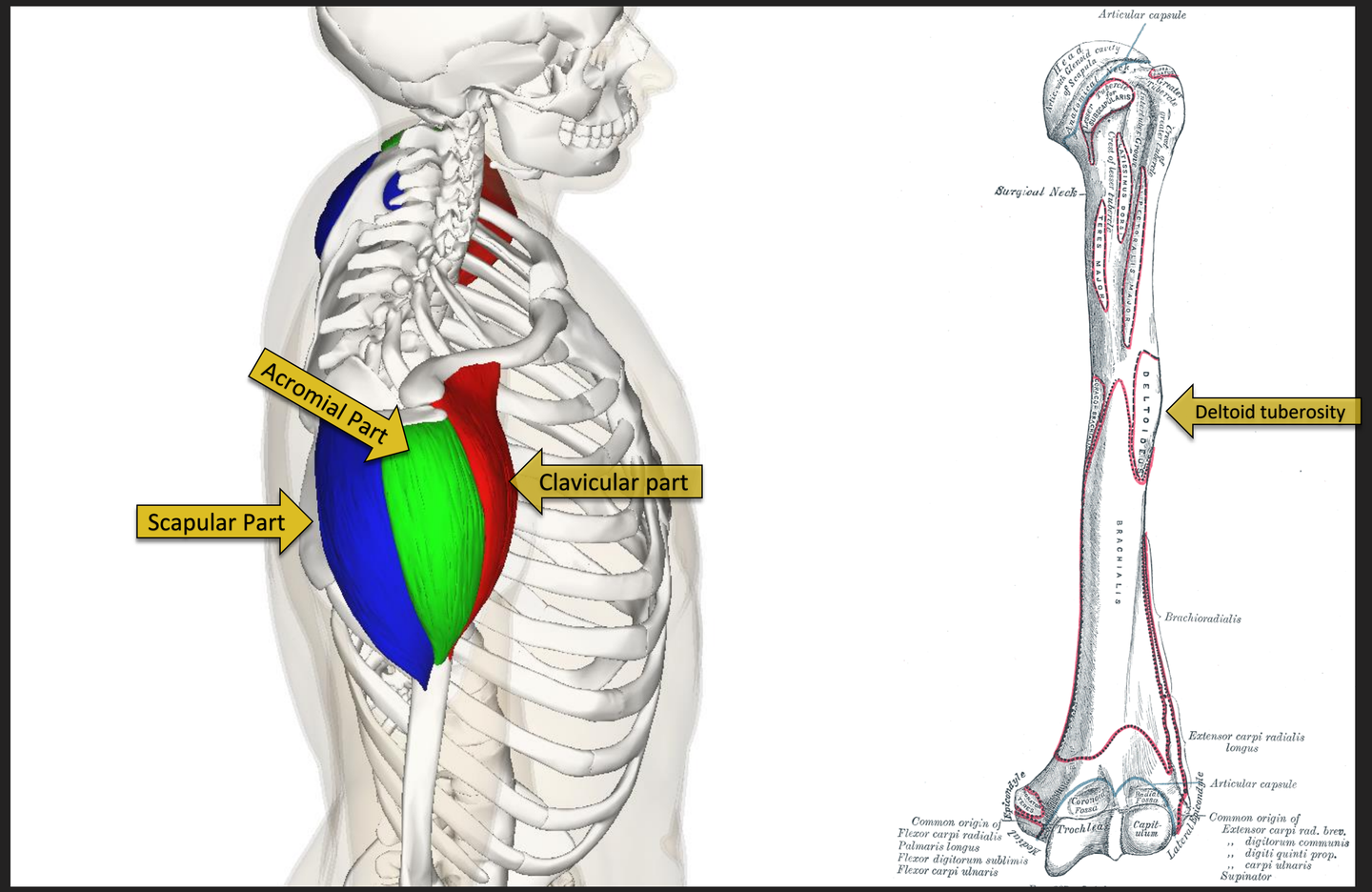


Deltoid muscle

Kratos's shoulder loses volume

Deltoid Muscle

- Anterior or clavicular part
- Posterior or scapular part
- Intermediate or acromial part



Side view of the deltoid muscle

Bone Mapping

Origin:

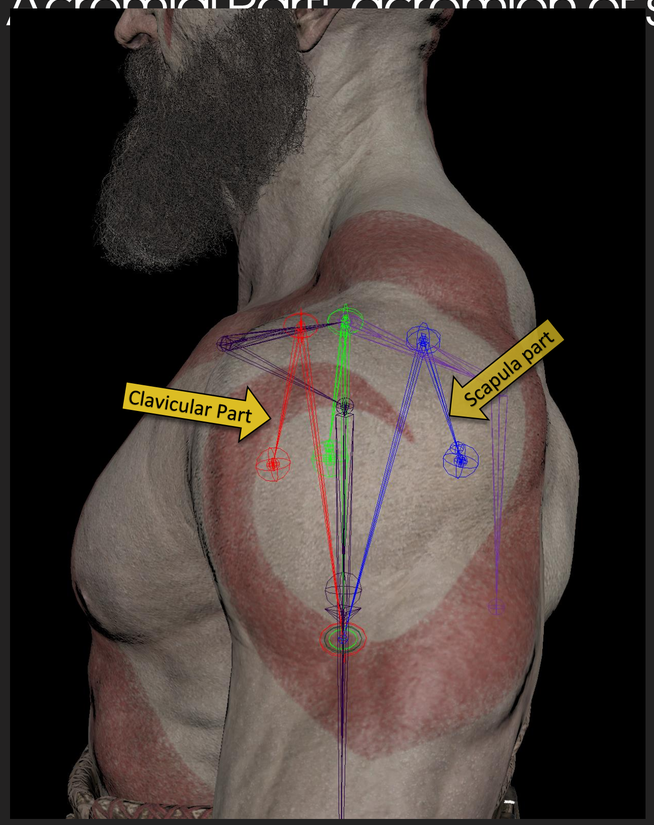
Clavicular Part: lateral third of clavicle

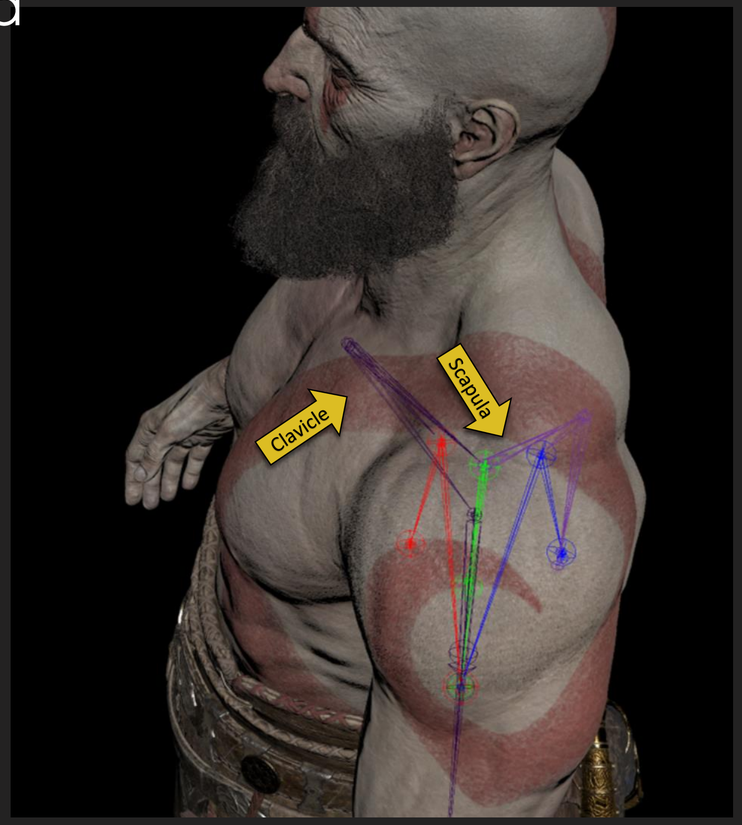
Scapular Part: lateral third of the spine of scapula

Acromial Part: acromion of scapula

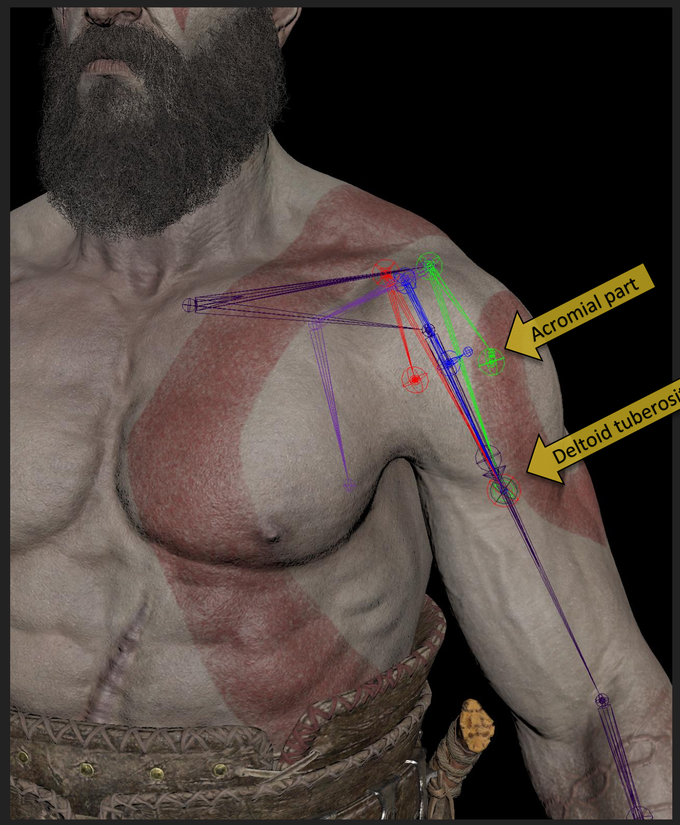
Insertion:

Deltoid tuberosity





Joint layout of deltoid muscle component

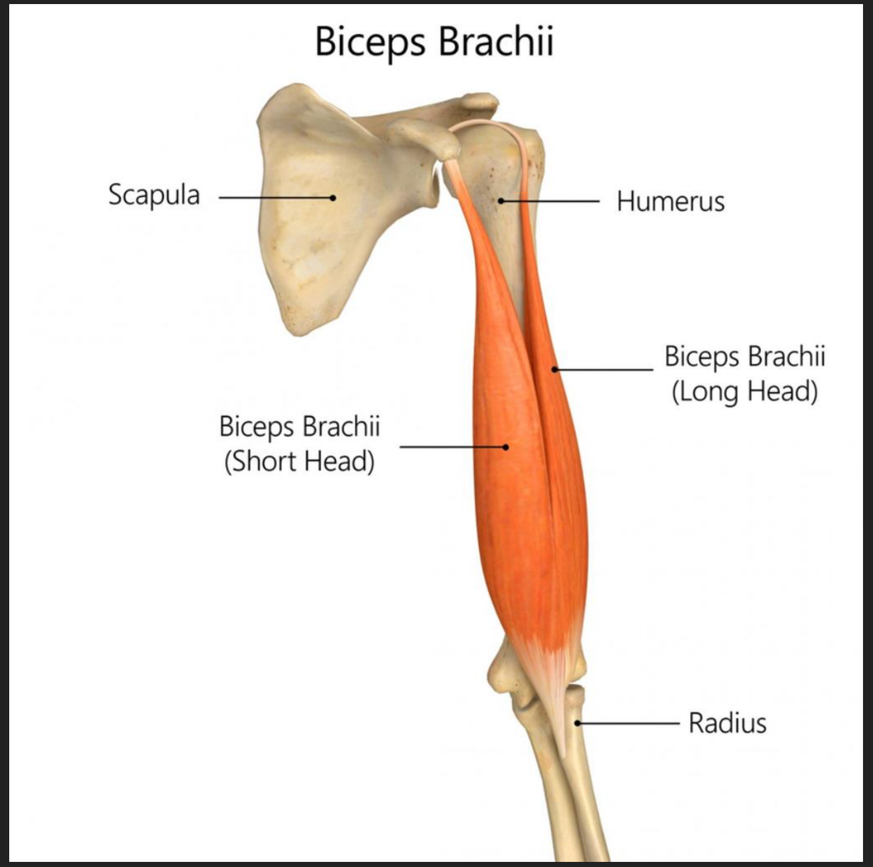


Biceps Brachii Muscle

Biceps Brachii Muscle

Origin: Scapula bone

Insertion: Radial tuberosity and the fascia of the forearm



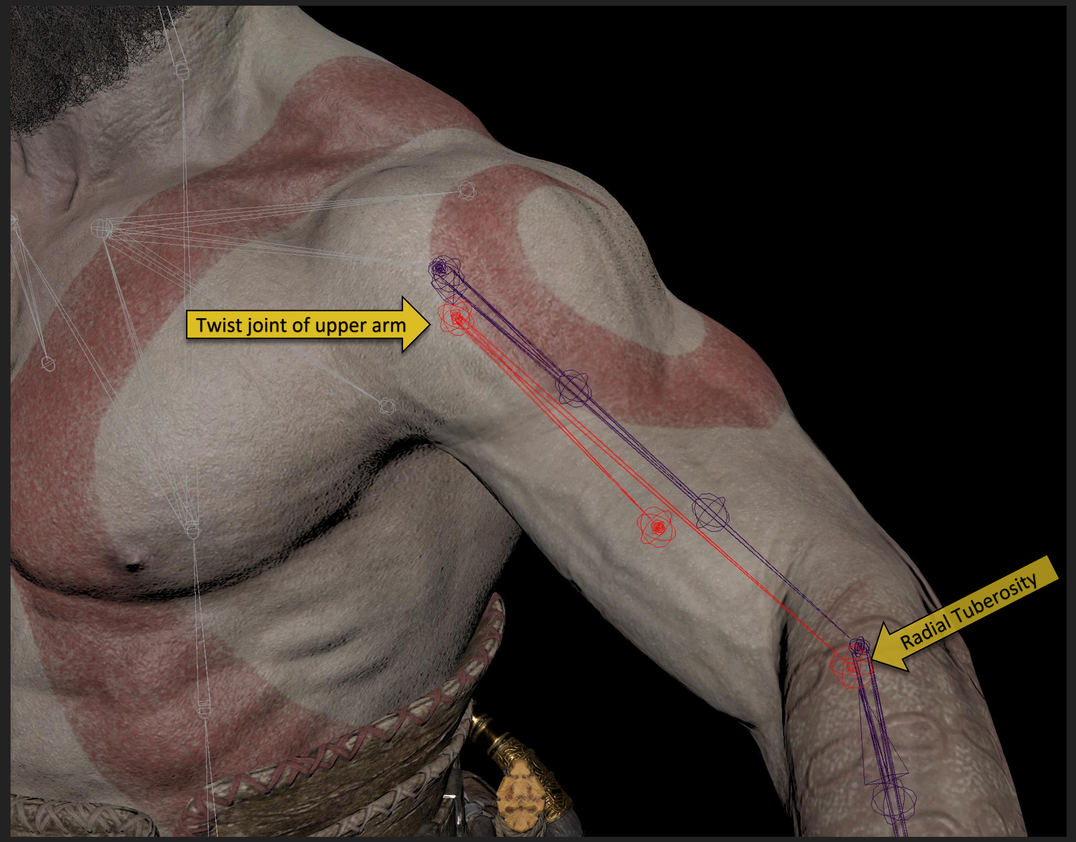
Bone Mapping

Origin:

Scapula bone

Insertion:

Radial tuberosity

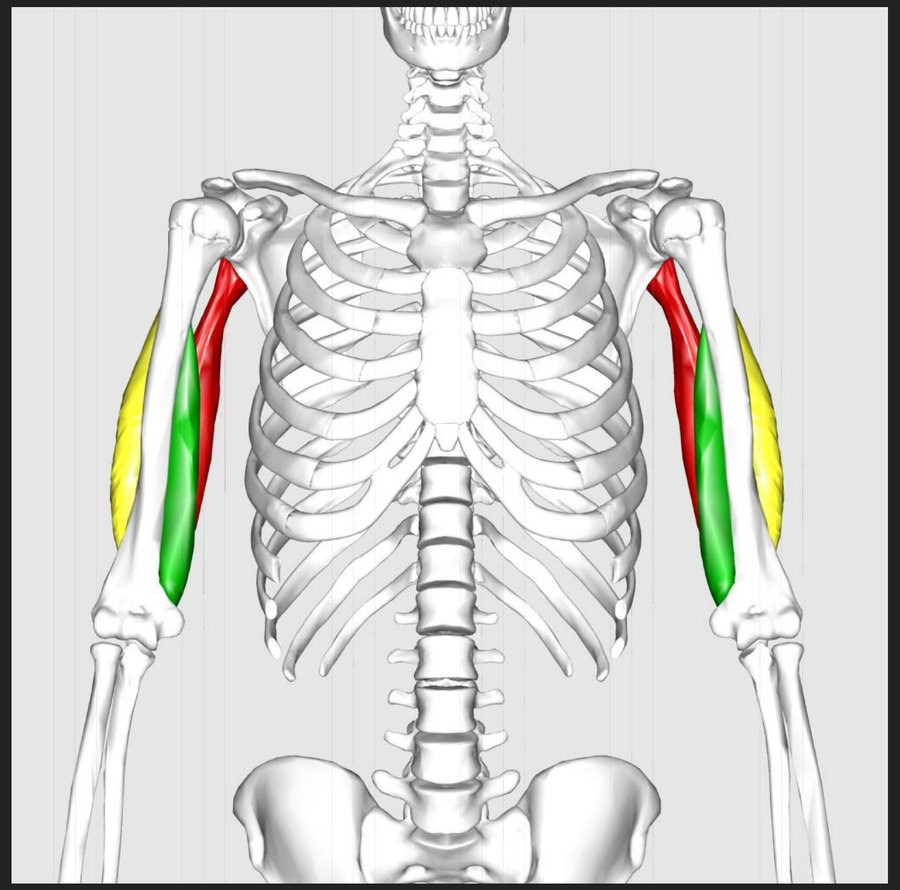


Joint layout of biceps muscle component

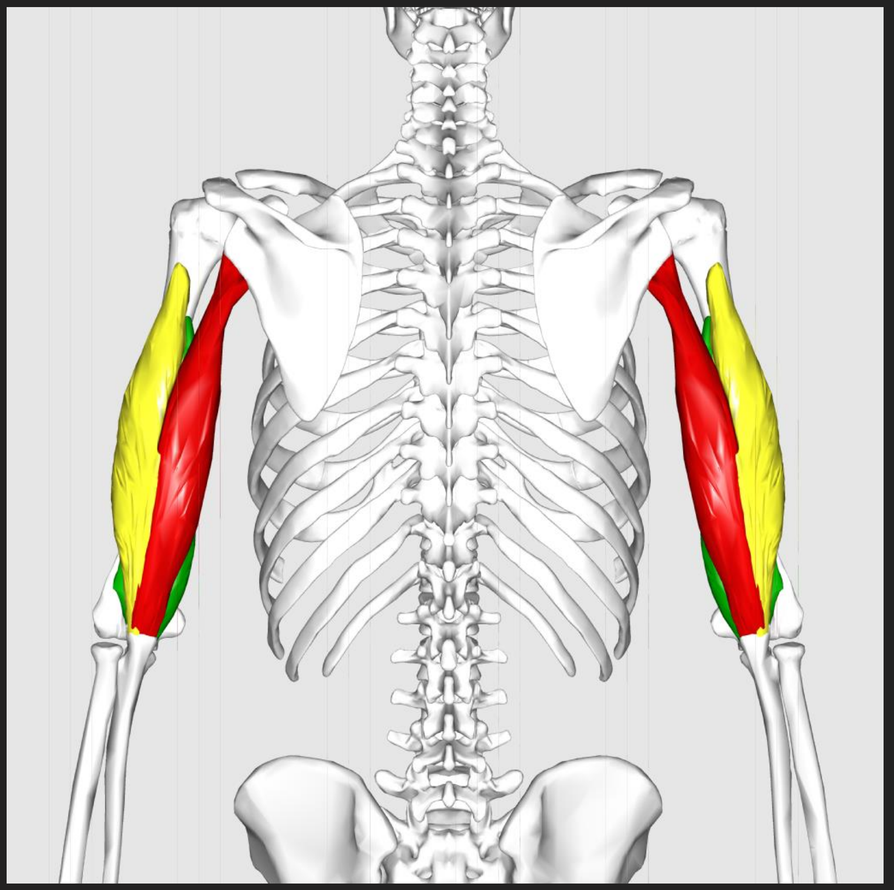
Forearm flexion, supination and pronation

Triceps Brachii Muscles

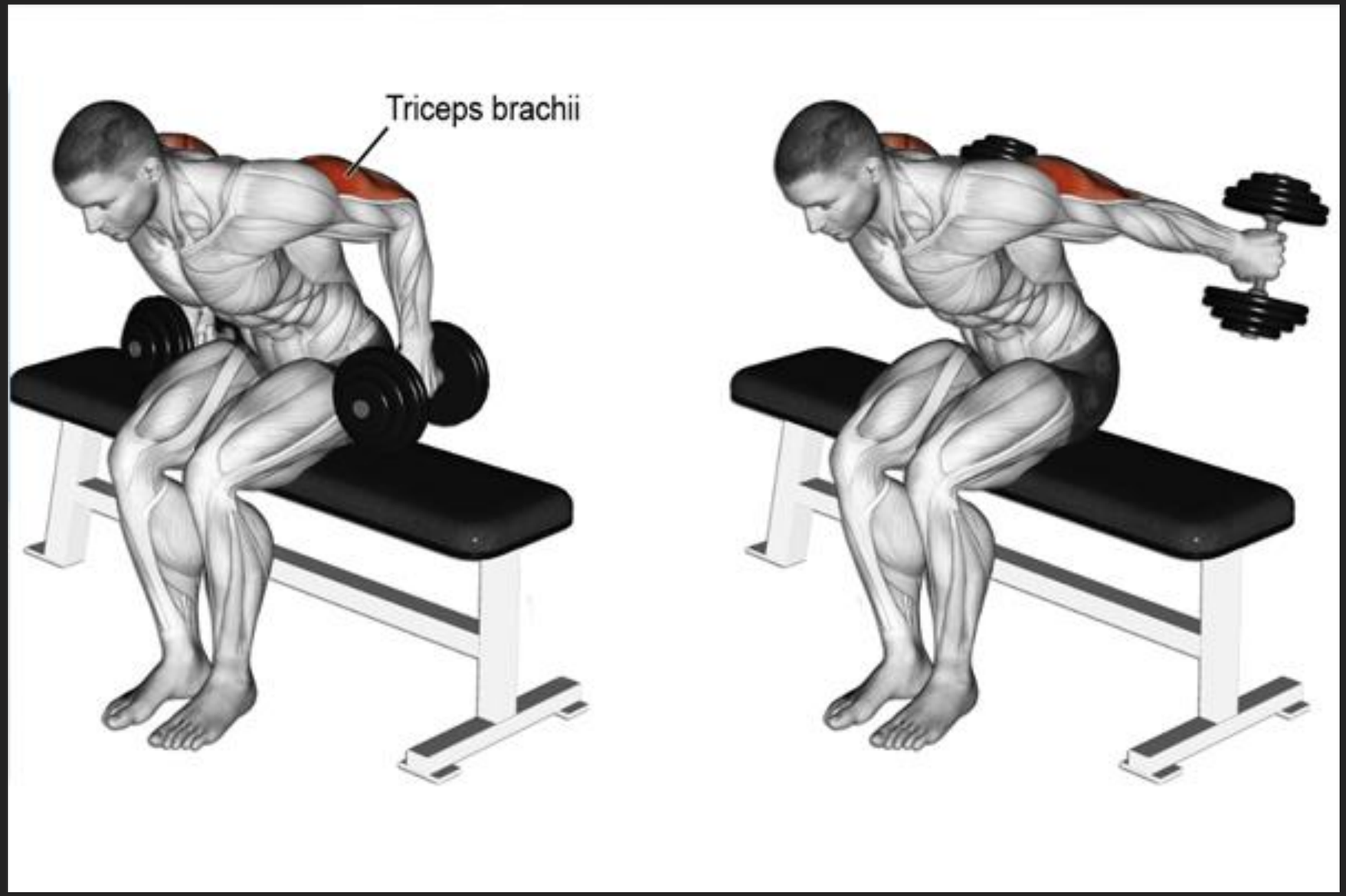
- Long head
- Lateral head
- Medial head



Anterior view



Back view


Triceps brachii

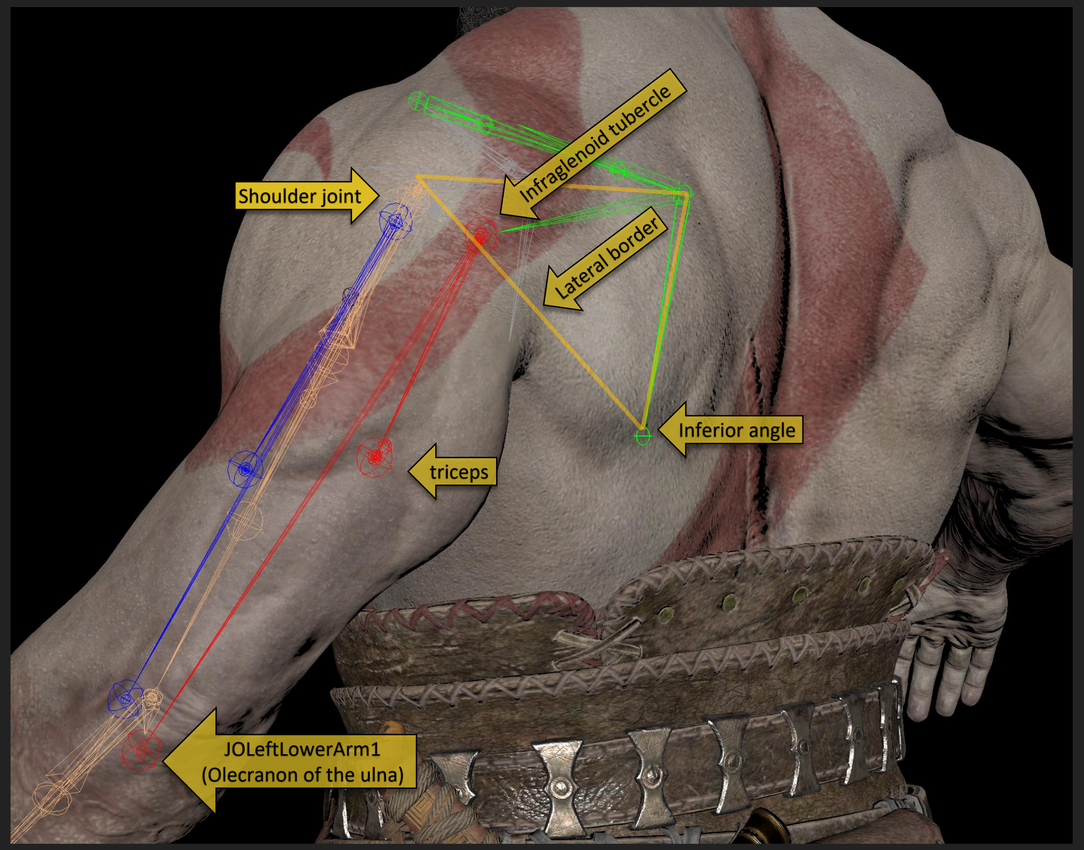
Bone Mapping

Origin:

Infraglenoid tubercle of the scapula

Insertion:

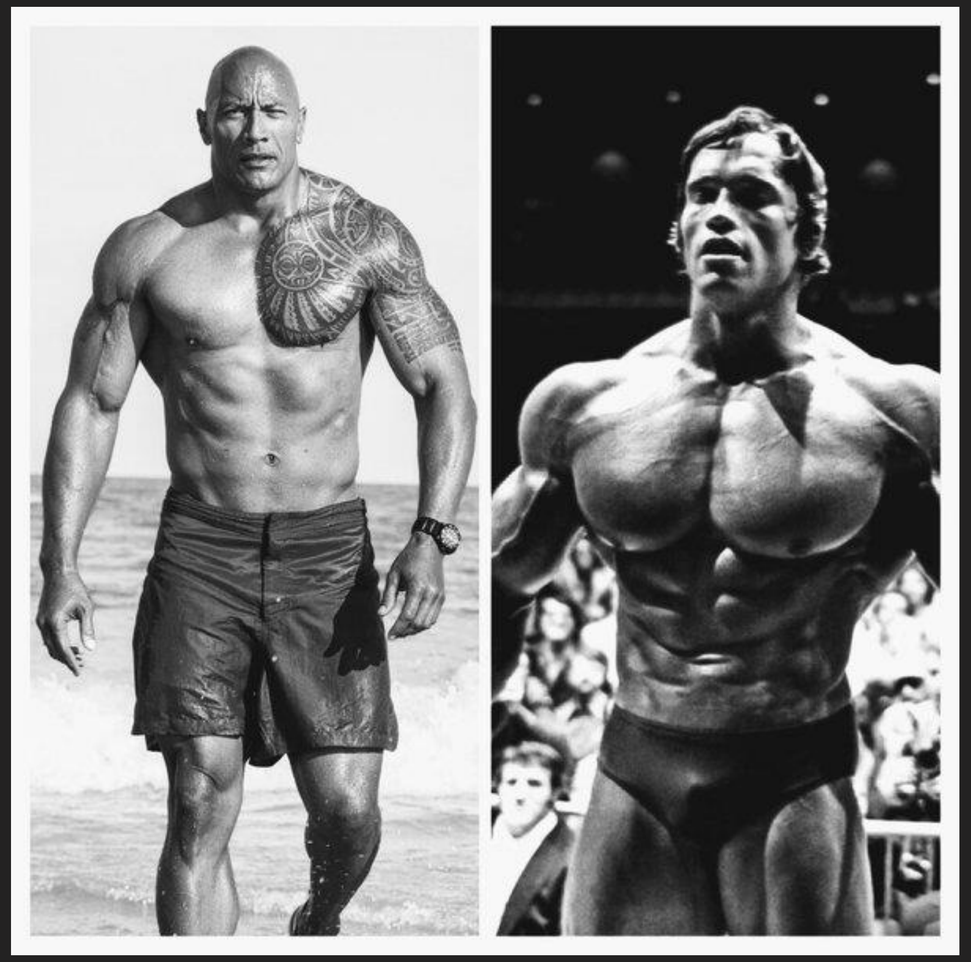
Olecranon process of ulna

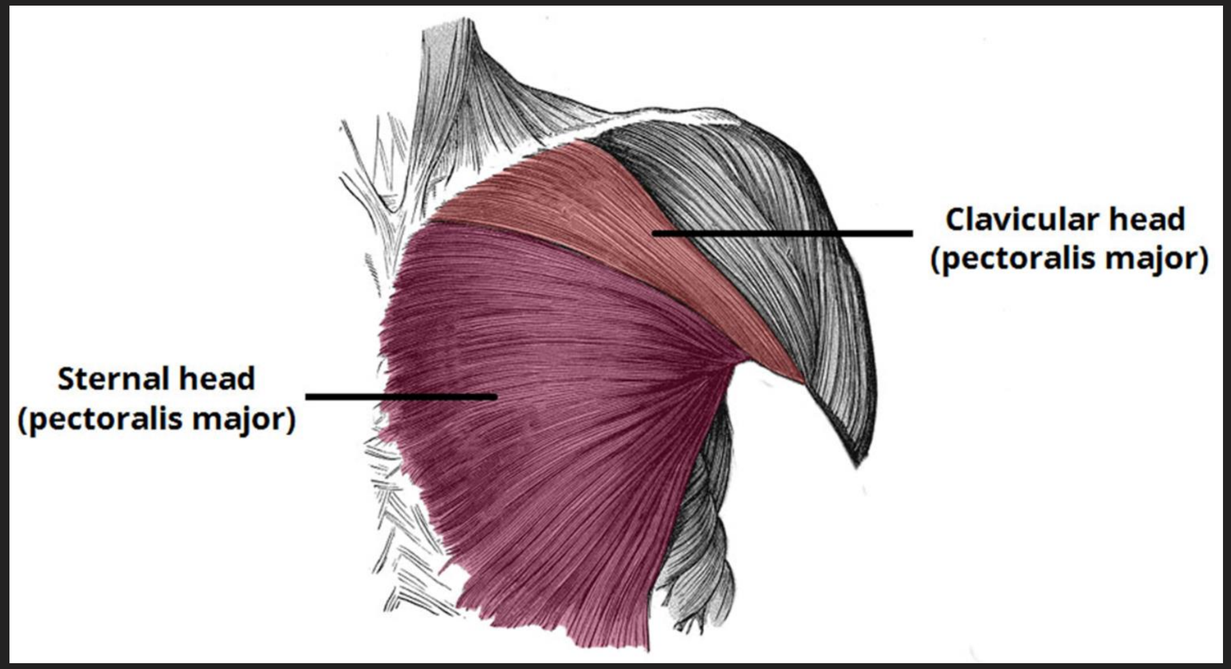


Joint layout of triceps muscle component

Forearm flexion and extension

Pectoralis Major Muscles

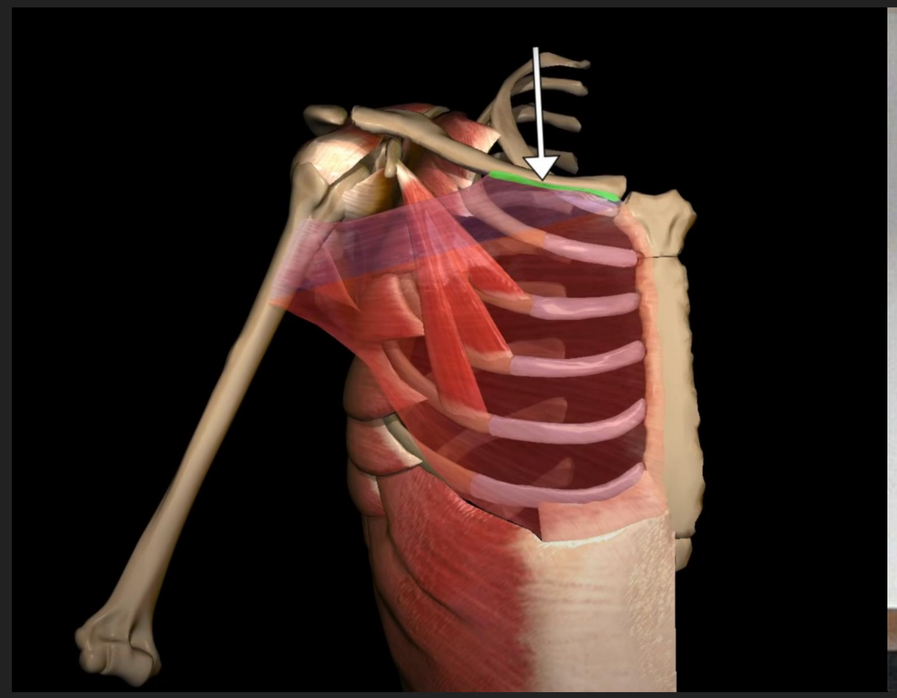




Clavicular Part

Origin: Medial half of the clavicle

Insertion: Intertubercular groove



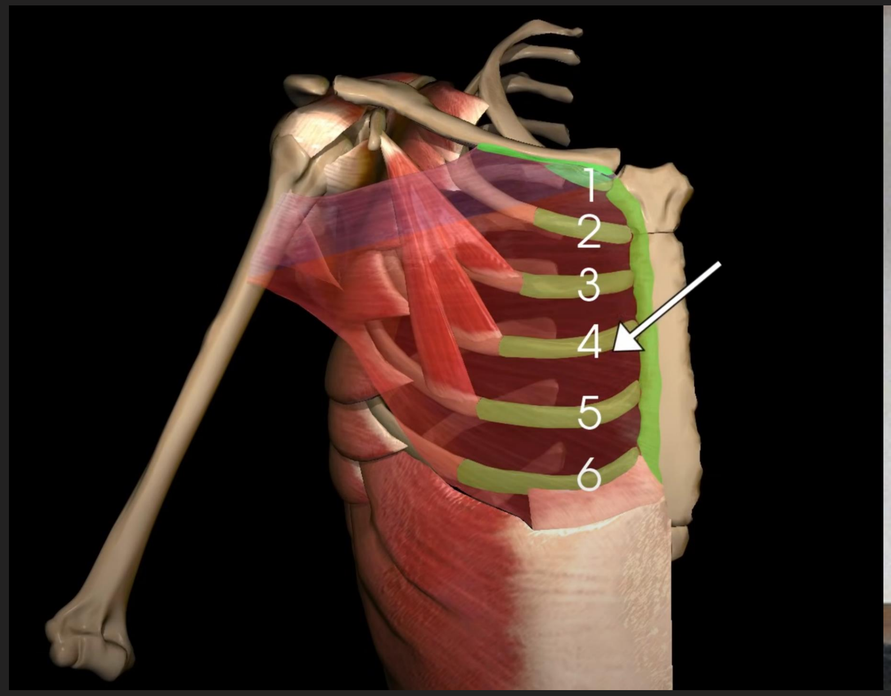
Origin of the clavicular part

Shoulder flexion

Sternocostal Part

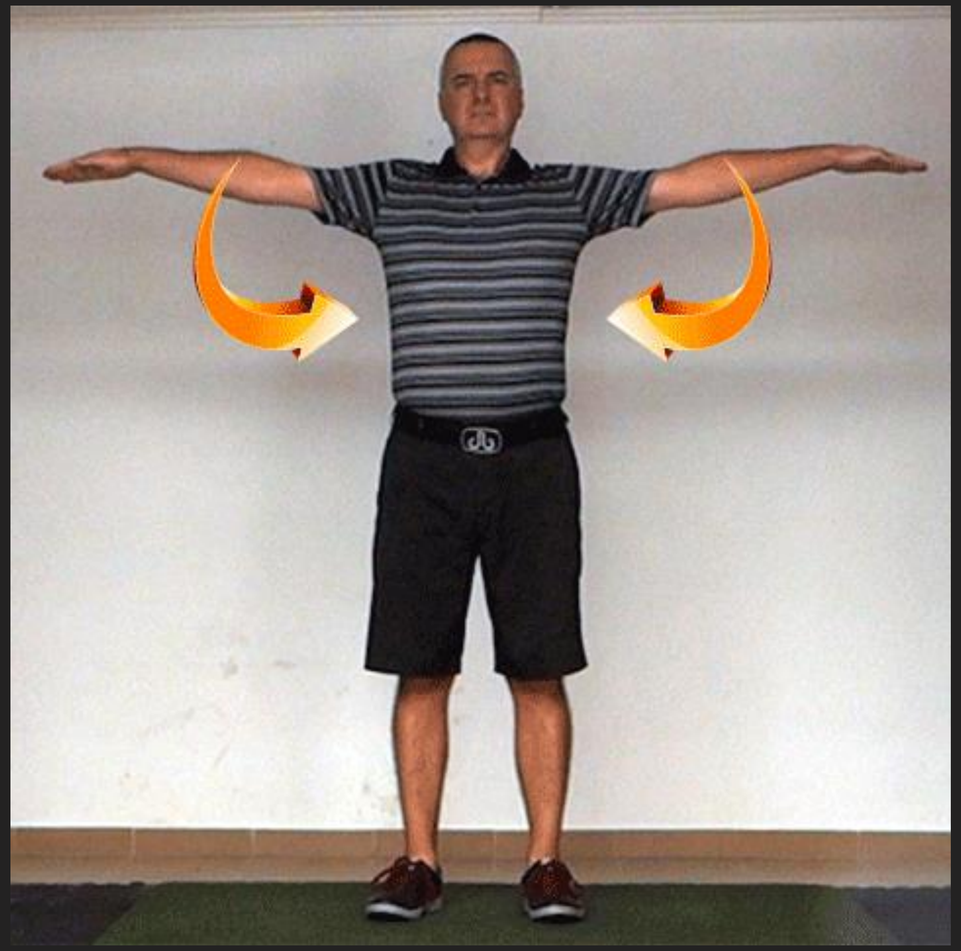
Origin: Anterior sternum and
Cartilages of ribs 1-6

Insertion: Intertubercular groove



Origin of the sternocostal part


Shoulder extension



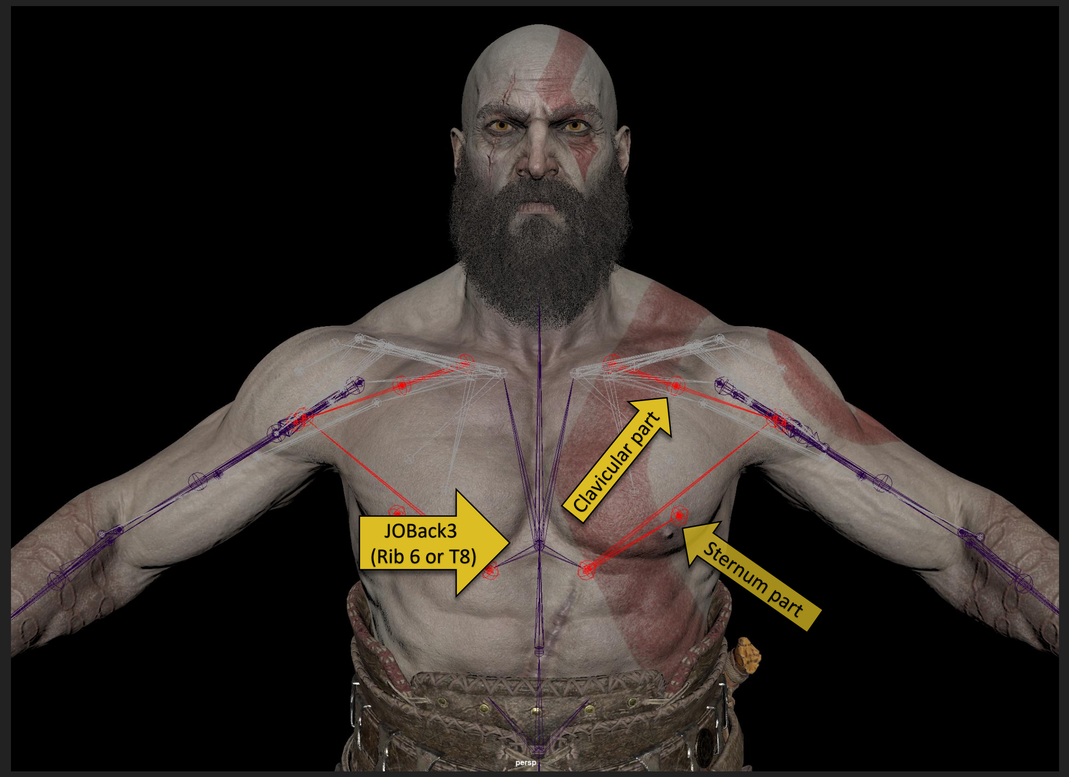
Shoulder adduction

Shoulder internal rotation

Bone Mapping

Origin:

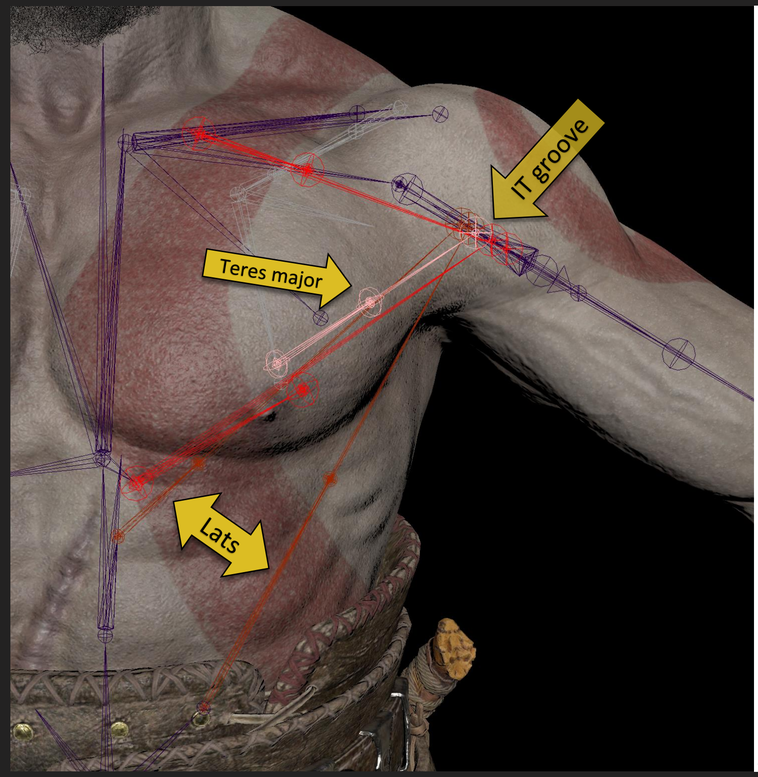
Clavicular Part: medial half the clavicle
Sternocostal Part: anterior surface of sternum,
costal cartilages of ribs 1-6

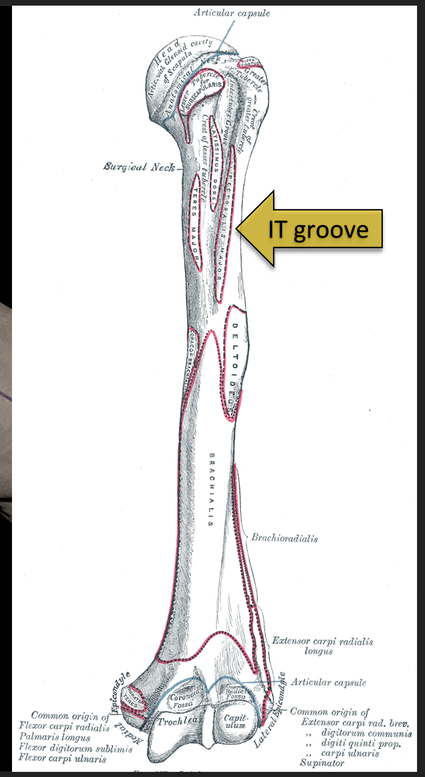


Joint layout of pecs muscle component

Insertion:

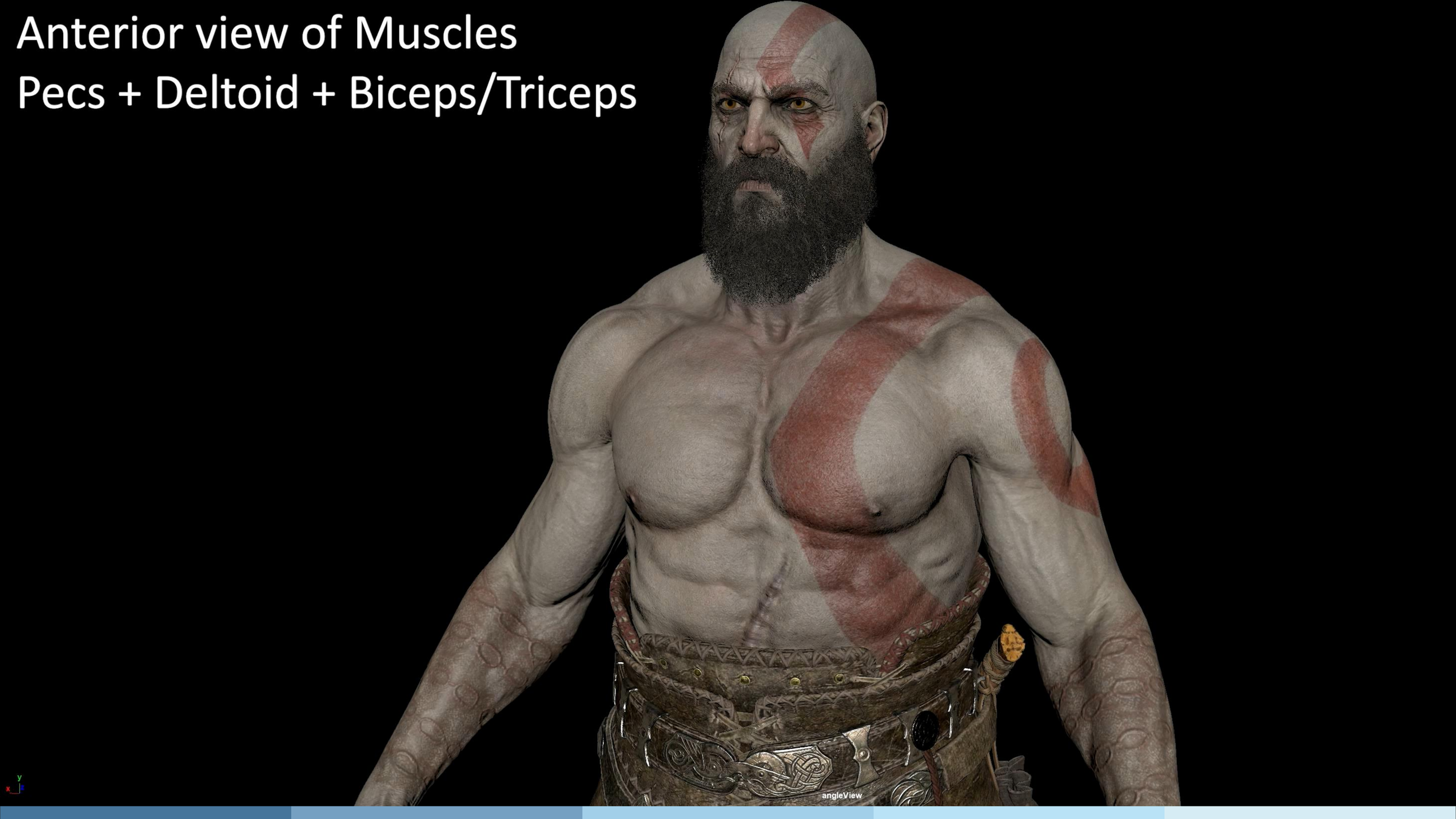
Intertubercular groove



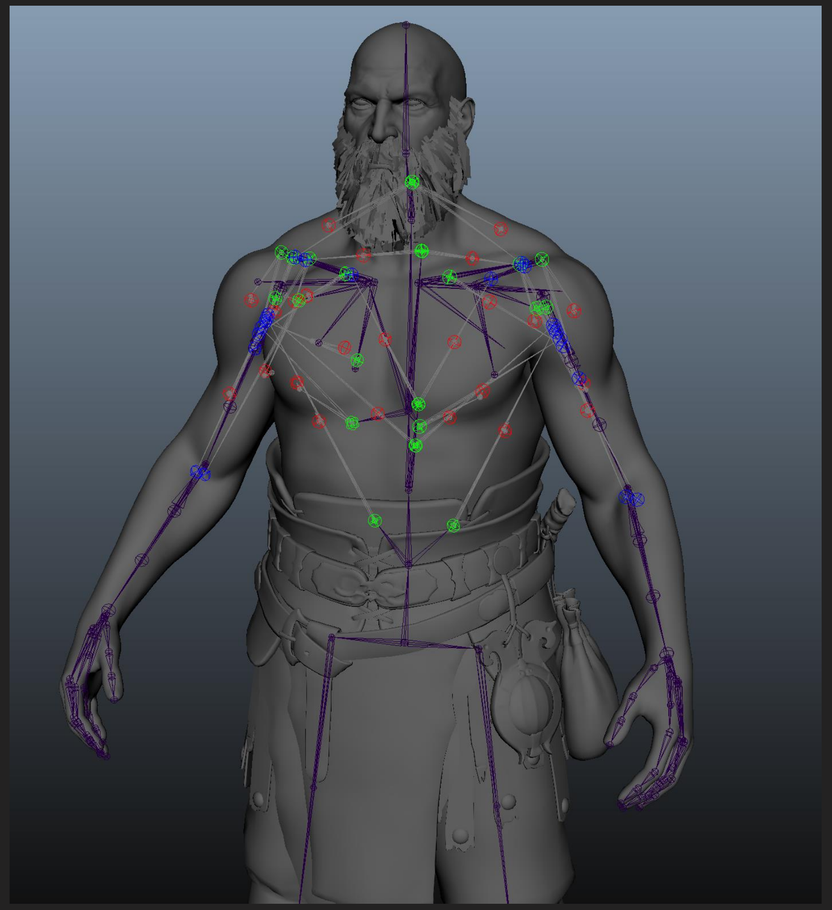


Common insertion for lats, teres major and pecs muscles

Anterior view of Muscles
Pecs + Deltoid + Biceps/Triceps



Import/Export Muscle Data



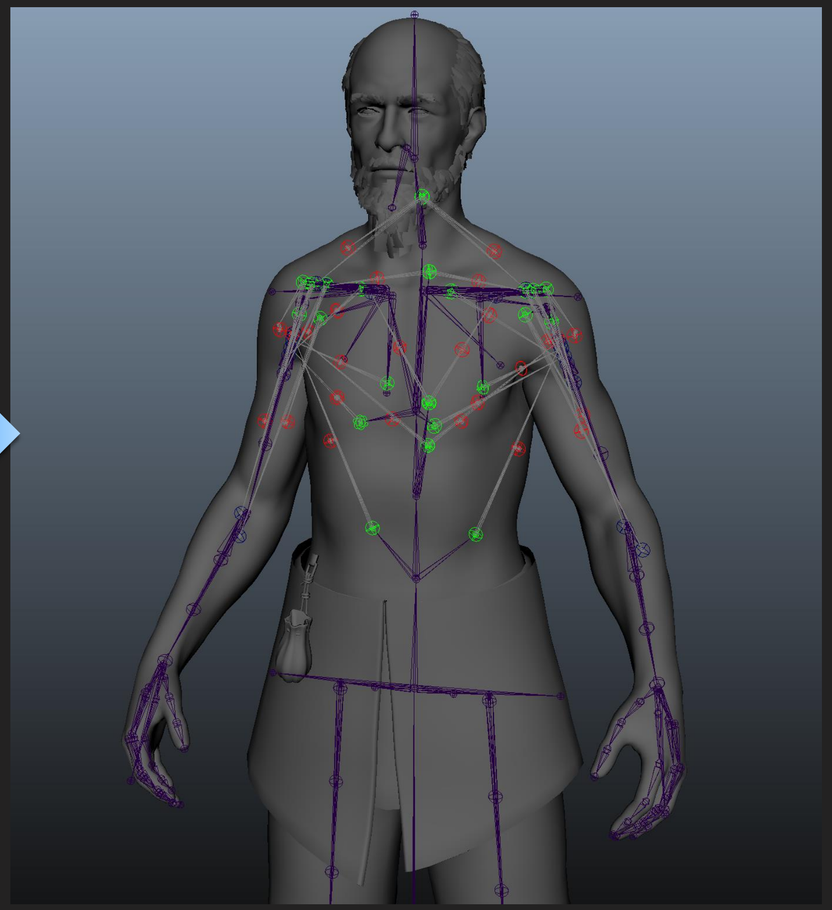
Kratos' muscle joint groups

EXPORT


```
{
    "Arms": {
        "Left": {
            "LeftBicep_muscleDriver": [
                -0.3358964044755173,
                1.473158021593473,
                -0.015958234332634685
            ],
            "LeftBicep_muscleInsertion": [
                -0.4492300414210402,
                1.4033466651324955,
                -0.02378082582593445
            ],
            "LeftBicep_muscleOrigin": [
                -0.20779553362803474,
                1.5548913560982407,
                0.01905507805450464
            ],
            "LeftTricep_muscleDriver": [
                -0.3142838437644579,
                1.4765397061817502,
                0.08780580607984975
            ],
            "LeftTricep_muscleInsertion": [
                -0.4737161834021363,
                1.3990106018449204,
                0.030882712286819457
            ],
            "LeftTricep_muscleOrigin": [
                -0.17771070486390236,
                1.5391032831016926,
                0.06494182061044583
            ]
        },
        "Right": {...}
    },
    "Deltoid": {...},
    "LatissimusDorsi": {...},
    "PectoralisMajor": {...},
    "TerasMajor": {...},
    "Trapezius": {...}
}
```


JSON serialization

IMPORT



Civilian's muscle joint groups

Increase Bone Influences per Vertex

Max 4 Influences

Max 10 Influences

God of War

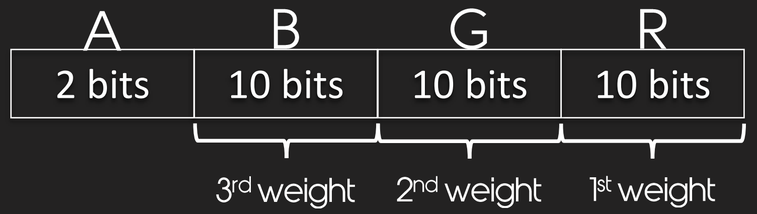
4 influences per vertex

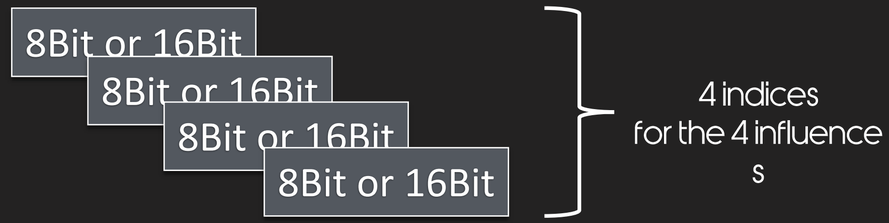
Weight Format: 32Bit_R10G10B10A_2_Unorm

Indices Format: 8Bit_Uint or 16Bit_Uint

Weight Precision: 0.001

Reconstruct final weight in shader





God of War: Ragnarok

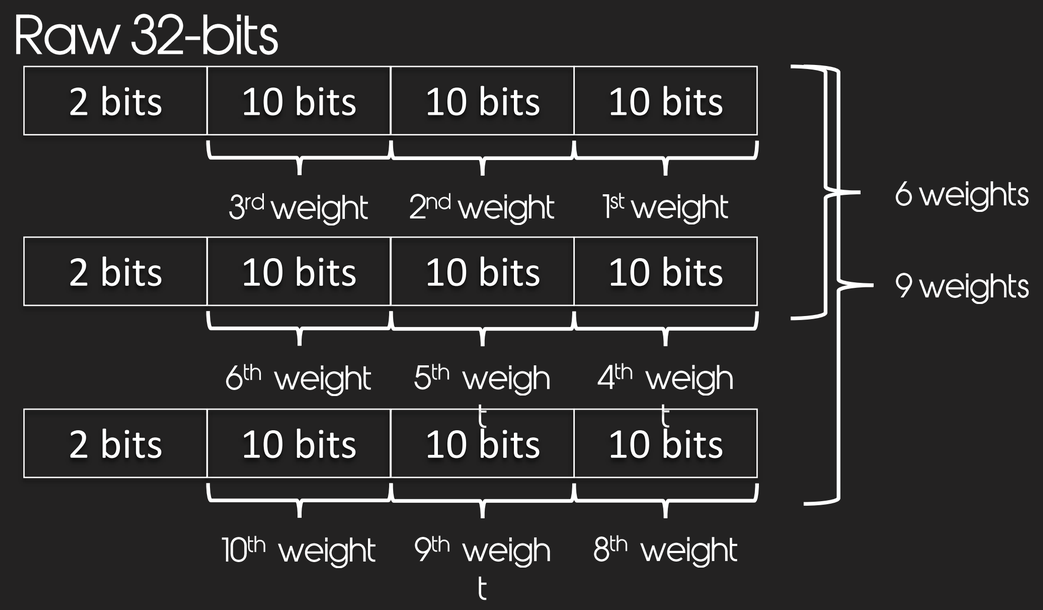
7/10 influences per vertex

Weight Format: 32Bit_Uint

Indices Format: 32Bit_Uint

Weight Precision: 0.001

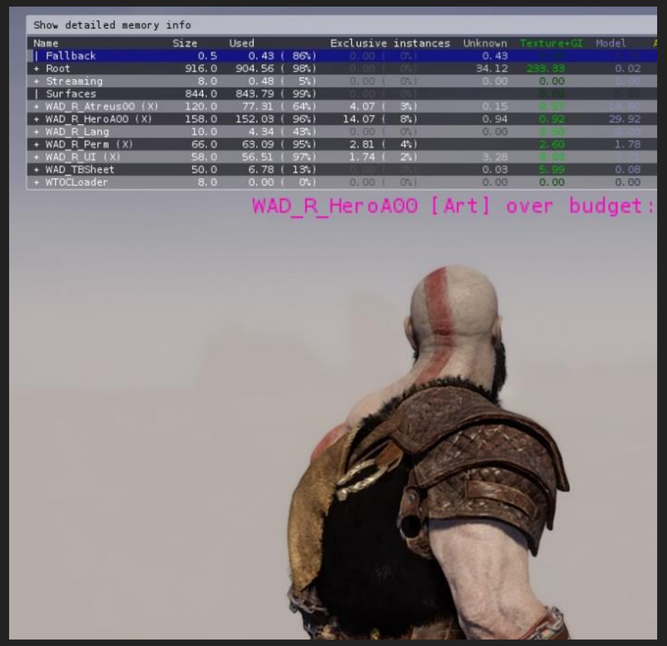
Reconstruct final weight in shader



Performance and Memory Difference

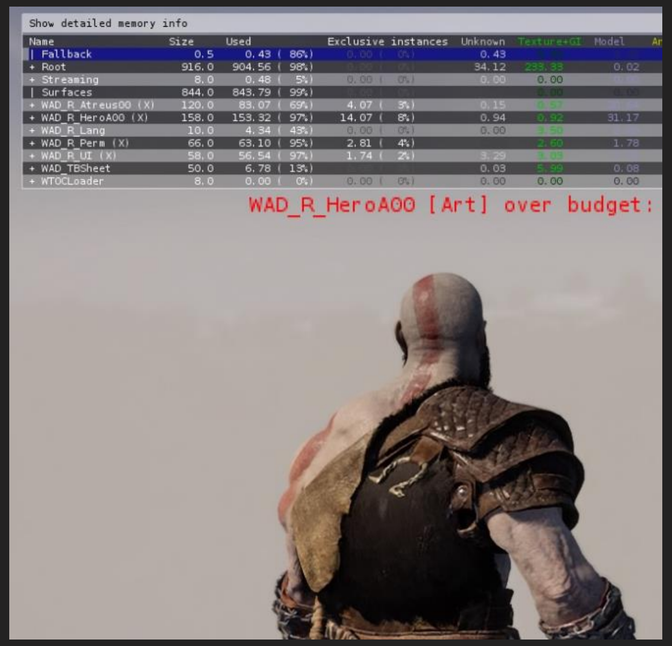
4 Influences:

Used Memory: 152.03
GPU Used: 36.51
CPU Used: 115.52
Depth: 0.36



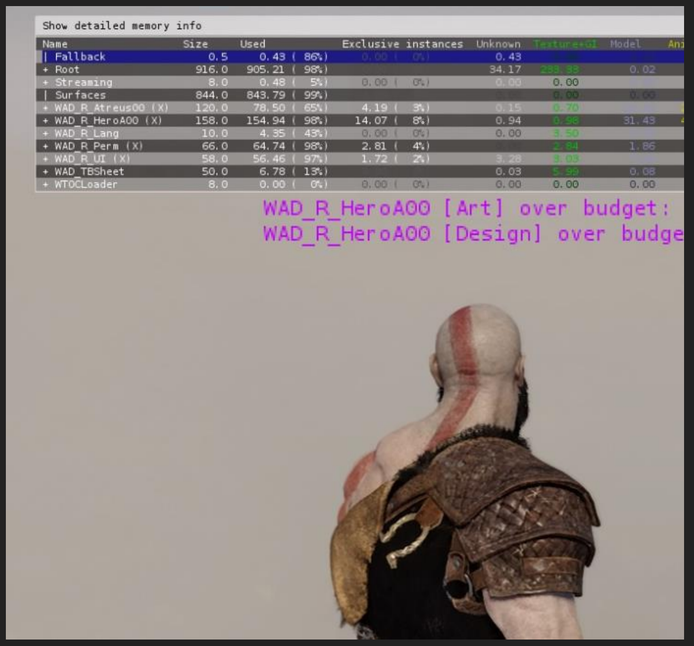
7 Influences:

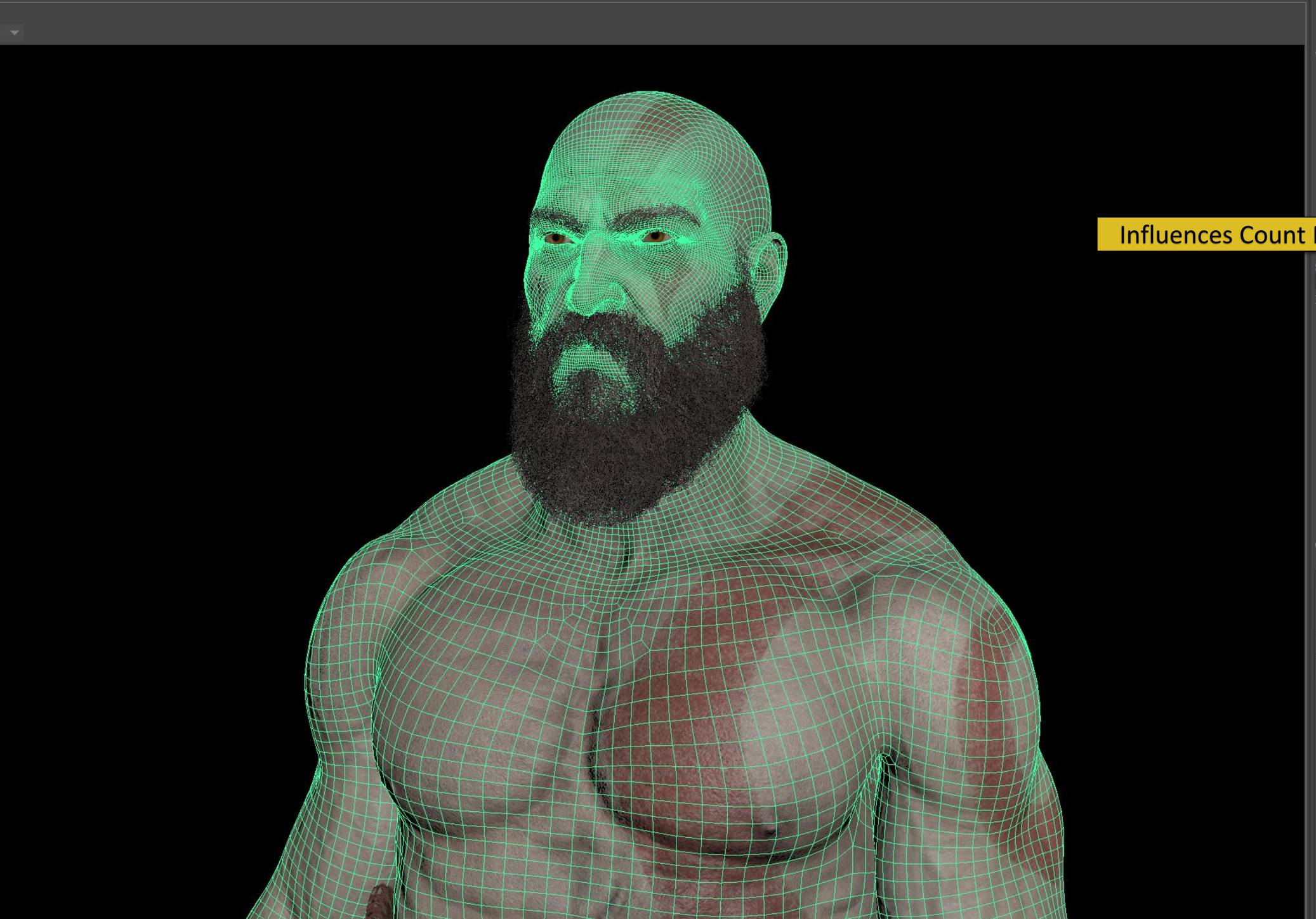
Used Memory: 153.32
GPU Used: 37.80
CPU Used: 115.52
Depth: 0.41



10 Influences:

Used Memory: 154.94
GPU Used: 38.27
CPU Used: 115.52
Depth: 0.46





Channels Edit Object Show

torso_head_mesh

Translate X	0
Translate Y	0
Translate Z	0
Rotate X	0
Rotate Y	0
Rotate Z	0
Scale X	1
Scale Y	1
Scale Z	1
Visibility	on
Influences Count Preset	10
Shadow Preset	NoShadow

Influences Count Flag

SHAPES
torso_head_meshShape
INPUTS
layer1
skinCluster154
neckHeadPSD
RightNose1_PSD
LeftNose1_PSD
RightOuterBrow1_PSD
LeftOuterBrow1_PSD
RightInnerBrow1_PSD
LeftInnerBrow1_PSD
RightLowerEye1_PSD
LeftLowerEye1_PSD
RightUpperEye1_PSD
LeftUpperEye1_PSD
kratos_psd_export_tweak763
OUTPUTS

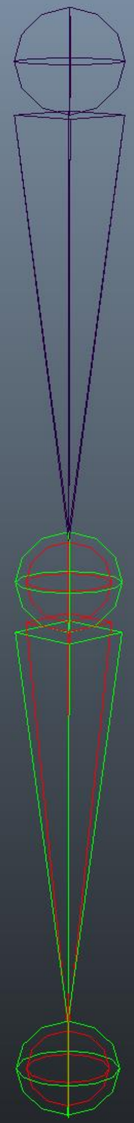
Display Anim

Layers Options Help

V	P		layer2
V	P		layer1
	P		HIDE_alManualShadowProxies
	P		_PSD
	P		_PoseReaders
	P		SoundEmitters
	P		AO_Proxies
	P		CollidableObject_Layer
	P		Decals1
V	P		_Cloth
	P	T	_ClothCollisions
	P		_Cloak
V	P		_Mesh
	P		_PermMesh


Jiggle Dynami
cs

Root Joint

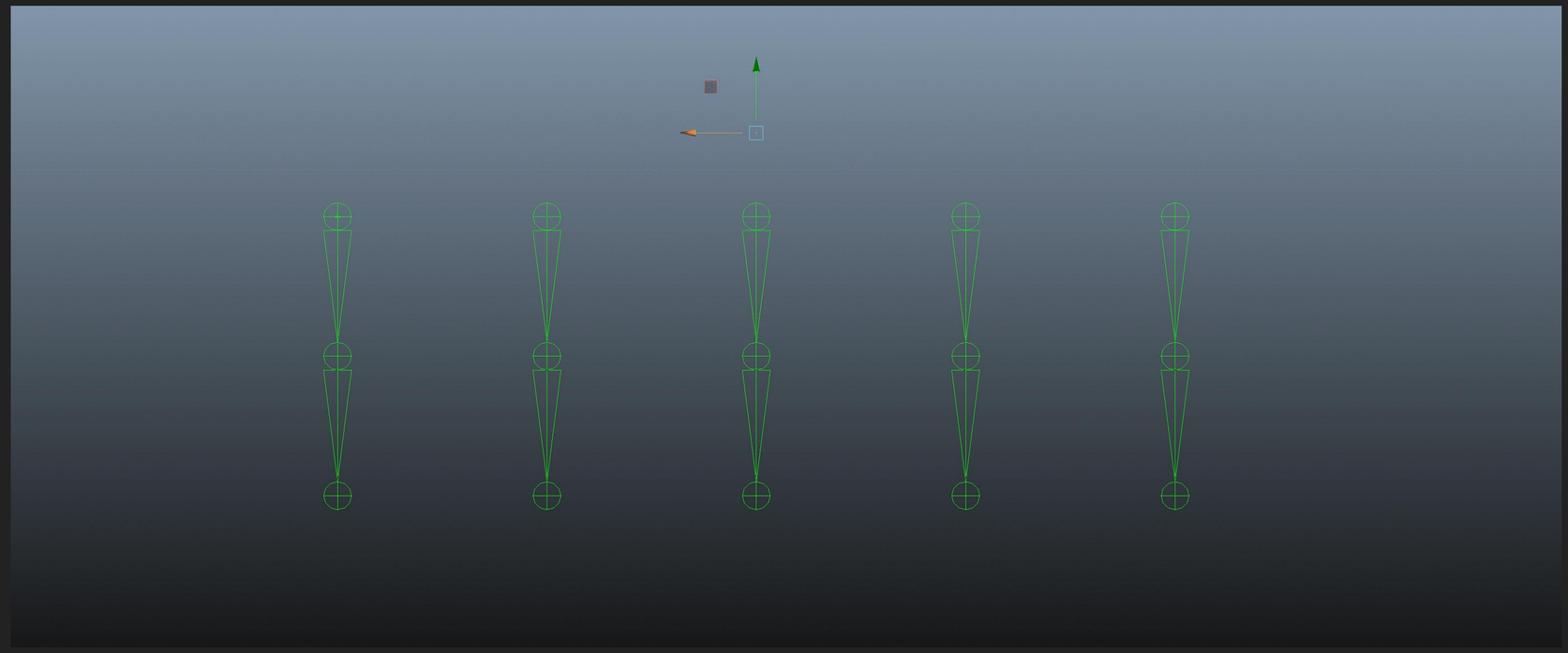


Pivot Joint

Jiggle Joint

Attributes

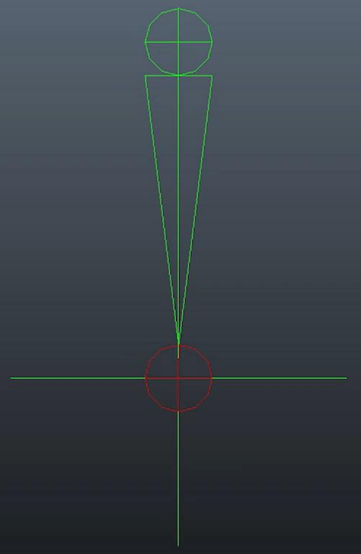
- Stiffness: range 0 – 1
- Damping: range 0 – 1
- ClampMajorAngle: range 0 – 90
- ClampMinorAngle: range 0 – 90



From left to right:

Stiffness: 0.05 Damping: 0.1 ClampMajorAngle: 90 ClampMinorAngle: 90	Stiffness: 0.15 Damping: 0.1 ClampMajorAngle: 90 ClampMinorAngle: 90	Stiffness: 0.15 Damping: 0.025 ClampMajorAngle: 90 ClampMinorAngle: 90	Stiffness: 0.1 Damping: 0.1 ClampMajorAngle: 90 ClampMinorAngle: 90	Stiffness: 0.1 Damping: 0.1 ClampMajorAngle: 15 ClampMinorAngle: 15

Implementation



Jiggle Point

Damped Spring Simple Motion

$$x = p - r$$

$$F = -\beta v - kx$$

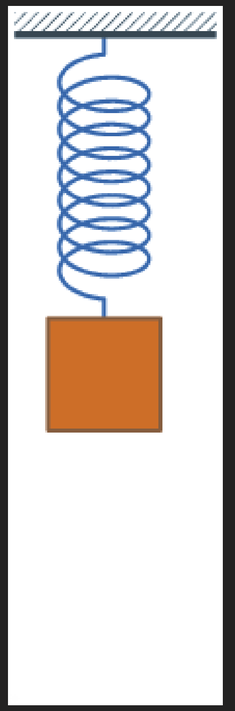
Semi-implicit Euler Integrator

$$v_{t+dt} = v_t + \mathrm{d}t \cdot stiffness \cdot (r - p) + dt \cdot damping \cdot v$$

$$x_{t+dt} = x_t + dt \cdot v_t$$

Ordinary Differential Equation (ODE)

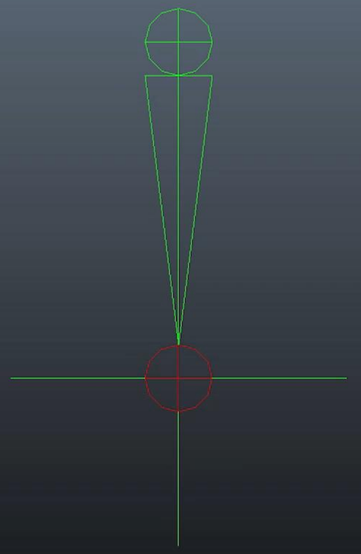
$$m\frac{d^2x}{dt^2} + \beta\frac{dx}{dt} + kx = 0$$



Related Articles:

- https://beltoforion.de/en/harmonic_oscillator/
- https://www.ryanjuckett.com/damped-springs/
- http://hyperphysics.phy-astr.gsu.edu/hbase/oscda.html

Implementation



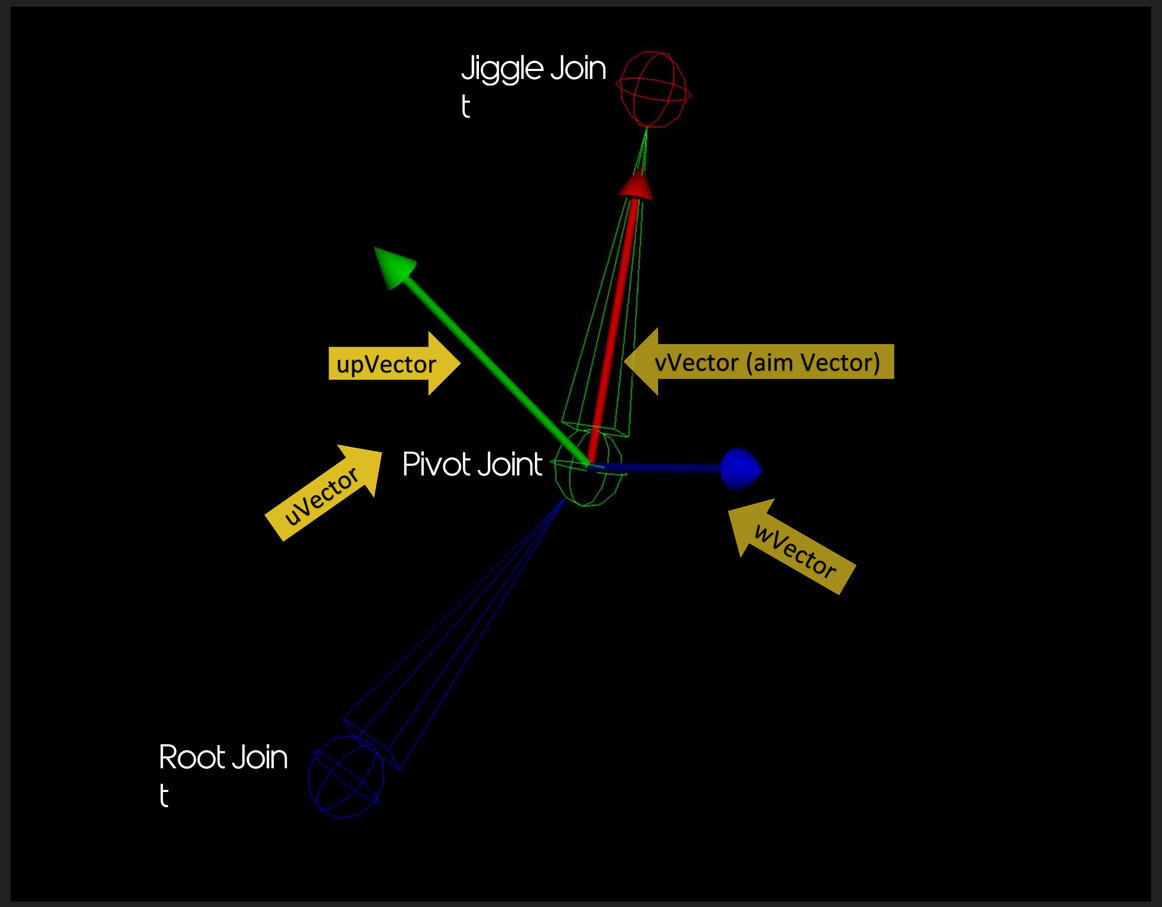
Aim Constraint

Aim Transformation

$$uVector = jigglePoint - PivotPoint$$
$$upVector = rootJoint.upAxis()$$
$$wVector = uVector \times upVecor$$
$$uVector = wVecor \times upVector$$

Rotation Matrix

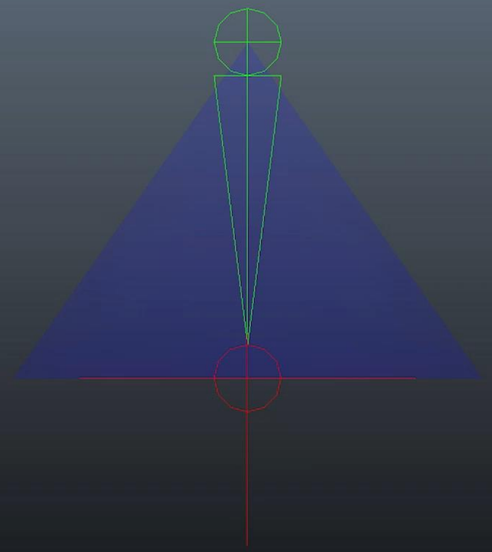
$$\begin{bmatrix} aimAxis \\ upAxis \\ aimAxis \times upAxis \end{bmatrix}^T \times \begin{bmatrix} uVector \\ vVector \\ wVector \end{bmatrix}$$



Matrix Decomposition

- http://eecs.qmul.ac.uk/~gslabaugh/publications/euler.pdf
- https://nghiaho.com/?page_id=846

Implementation



Cone Constrai
nt

Line – Elliptical Cone Intersection

Equation for elliptical cone

$$\frac{y^2}{h^2} = \frac{x^2}{a^2} + \frac{z^2}{b^2} \quad \Rightarrow \quad y^2 = \frac{x^2}{\tan^2 \alpha} + \frac{z^2}{\tan^2 \beta}$$

Half major angle at the vertex: $\alpha = \arctan \frac{a}{h}$

Half minor angle at the vertex: $\beta = \arctan \frac{b}{h}$

Equation for a line segment

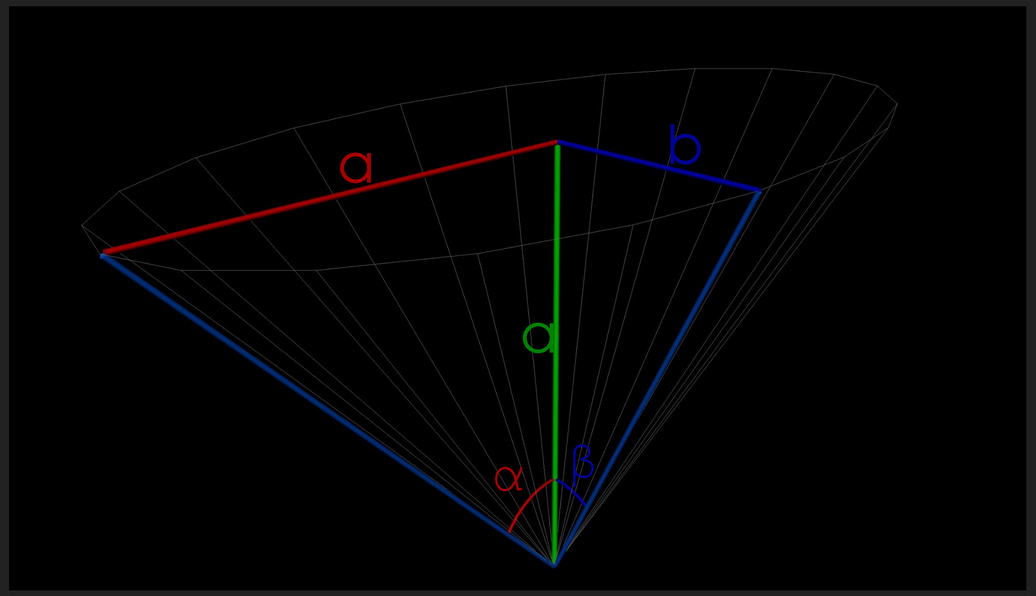
$$x = tx_1 + (1-t)x_0$$
$$y = ty_1 + (1-t)y_0 \qquad where\ t \in [0,1]$$
$$z = tz_1 + (1-t)z_0$$

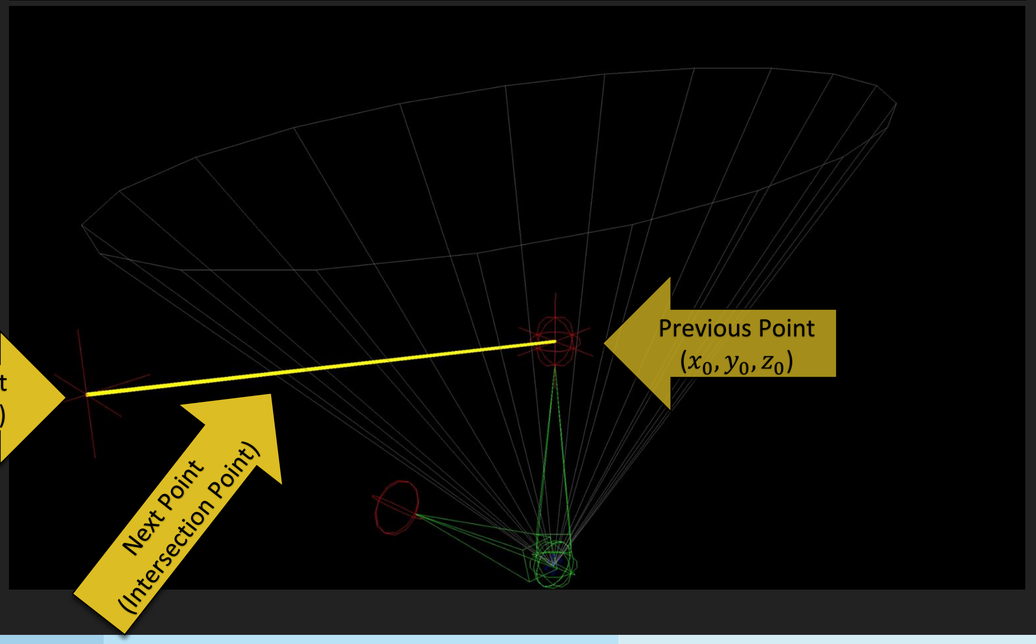
The form of a quadratic formula

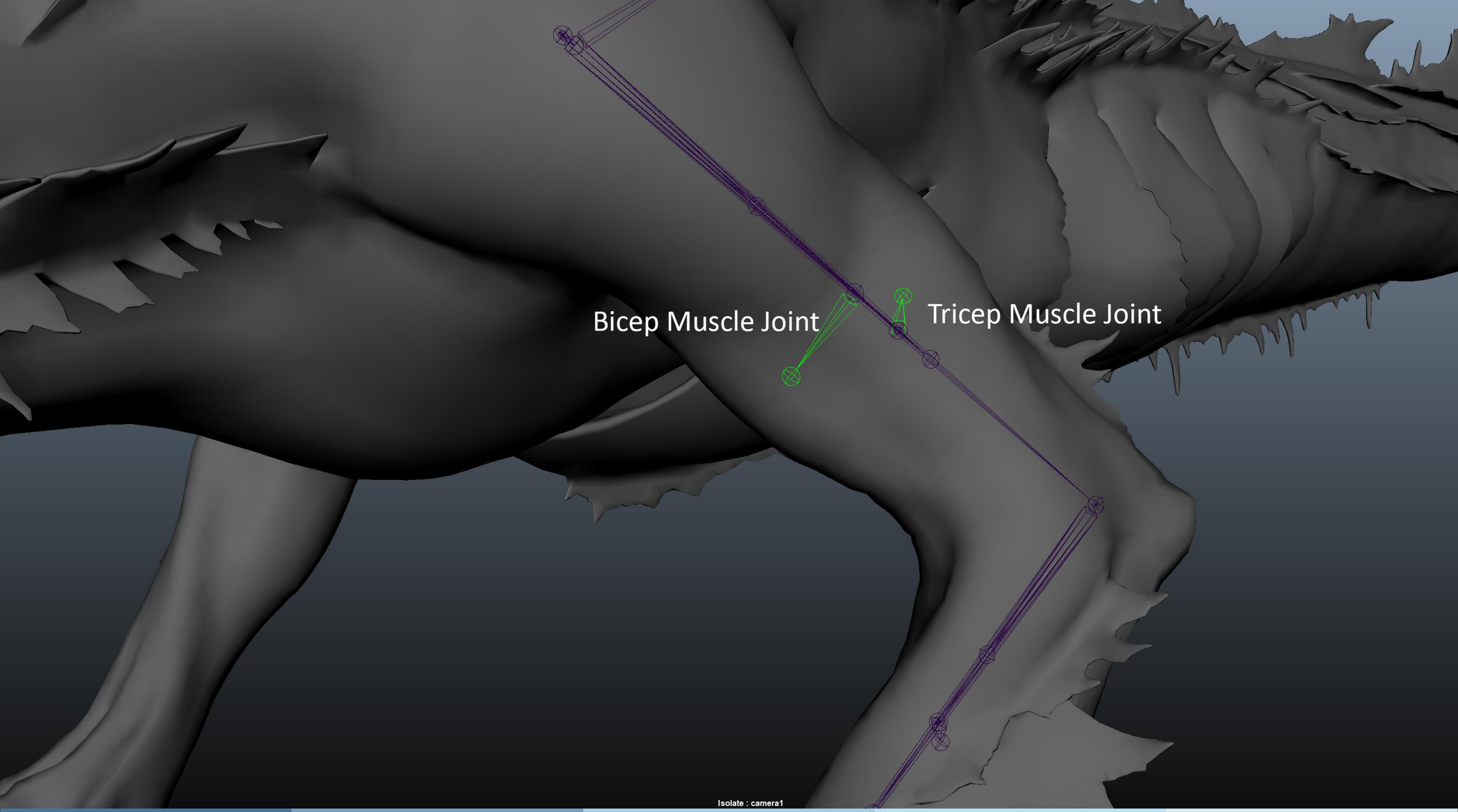
$$At^2 + Bt + C = 0$$

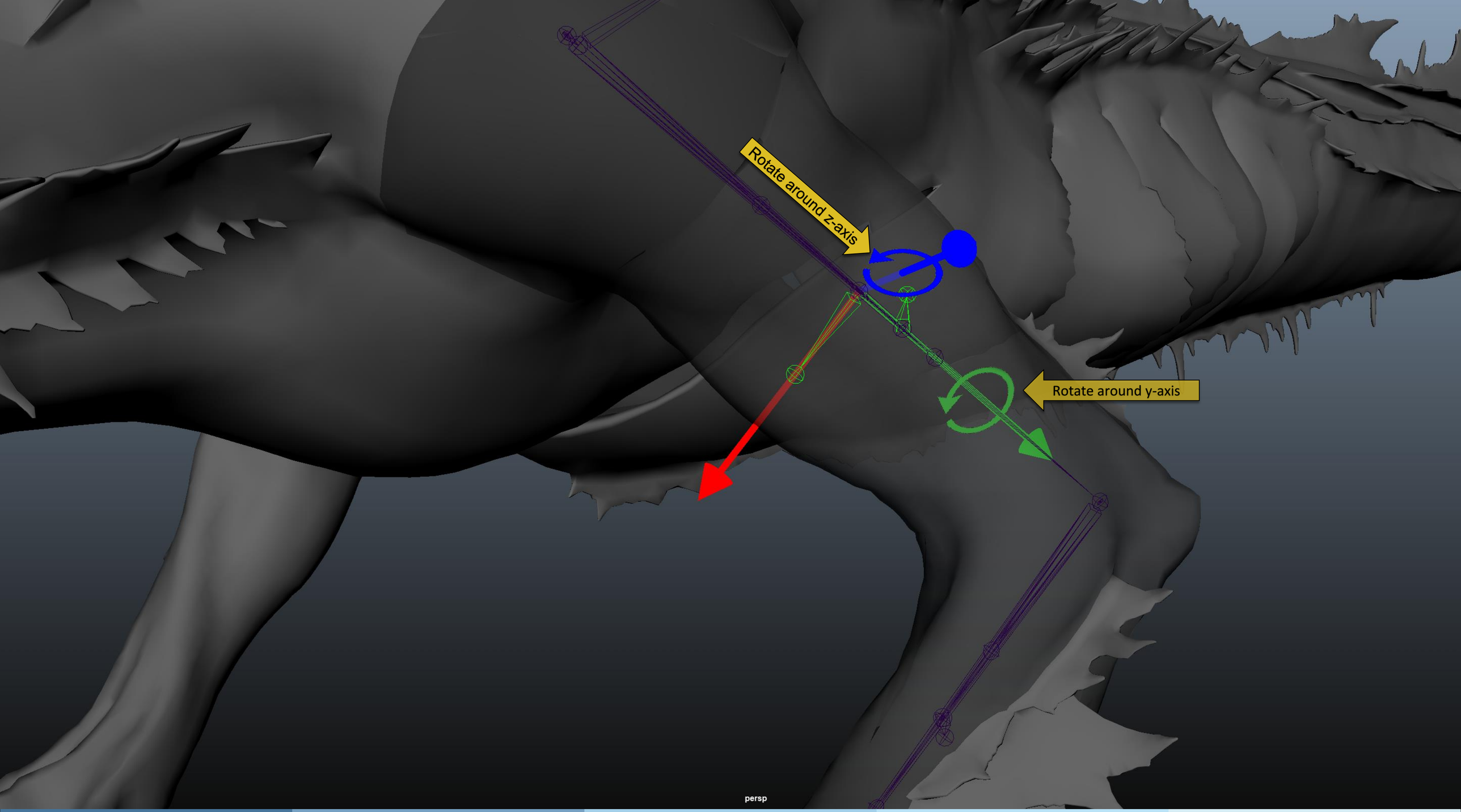
$where$

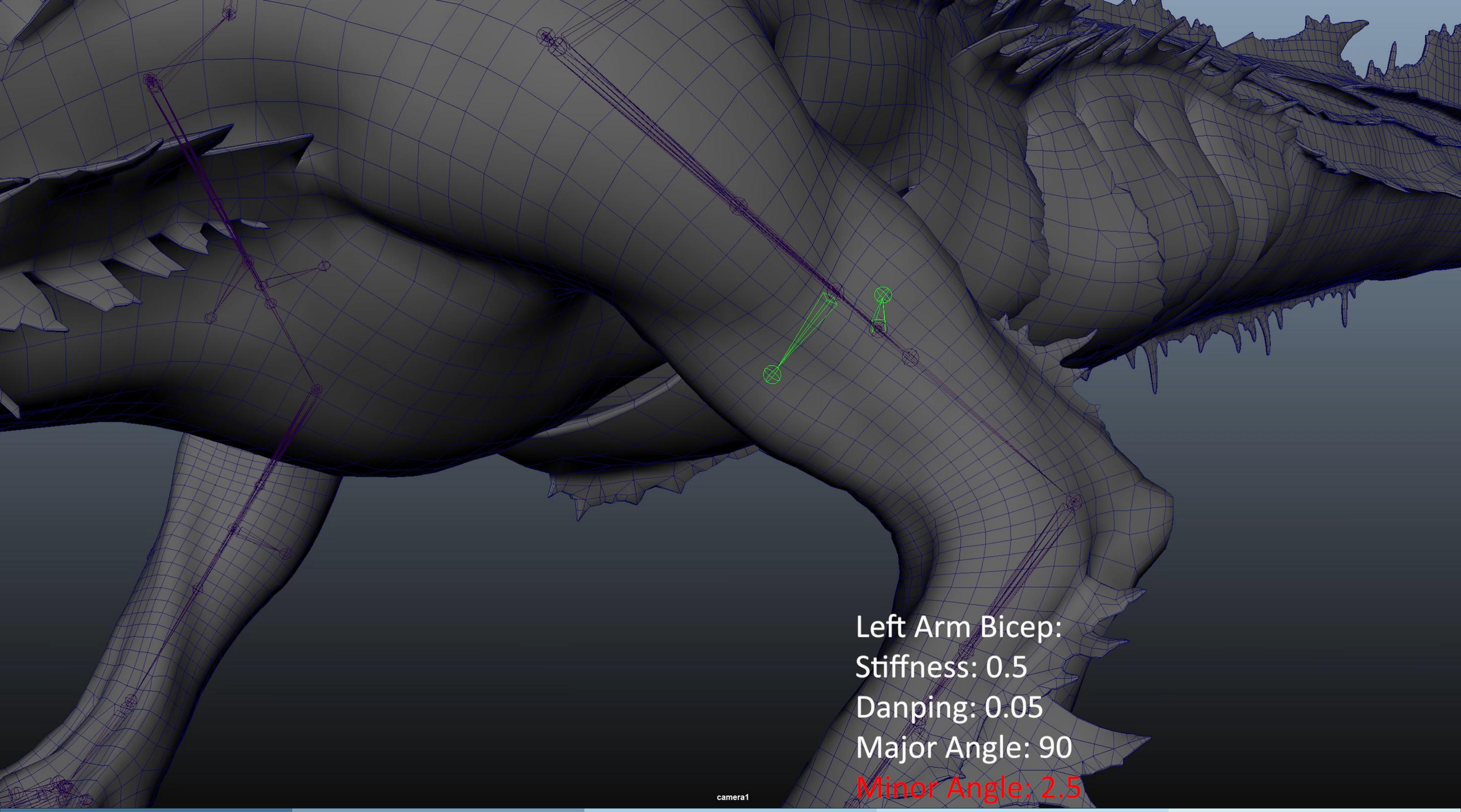
- $A = \frac{(x_1-x_0)^2}{\tan^2 \alpha} + \frac{(z_1-z_0)^2}{\tan^2 \beta} - (y_1 - y_0)^2$
- $B = \frac{2(x_1-x_0)x_0}{\tan^2 \alpha} + \frac{2(z_1-z_0)z_0}{\tan^2 \beta} - 2(y_1 - y_0)y_0$
- $C = \frac{x_0^2}{\tan^2 \alpha} + \frac{z_0^2}{\tan^2 \beta} - y_0^2$






Bicep Muscle Joint
Tricep Muscle Joint
Isolate : camera1


Rotate around z-axis
Rotate around y-axis
persp


Left Arm Bicep:
Stiffness: 0.5
Danping: 0.05
Major Angle: 90
Minor Angle: 2.5
camera1

Animation Reaction

Without Jiggle Bones

Bear body jiggle

Land dragon legs jiggle

Nidhogg head fins and arm jiggle

Yak ears jiggle

Combat Reaction

Thor’s hit animations

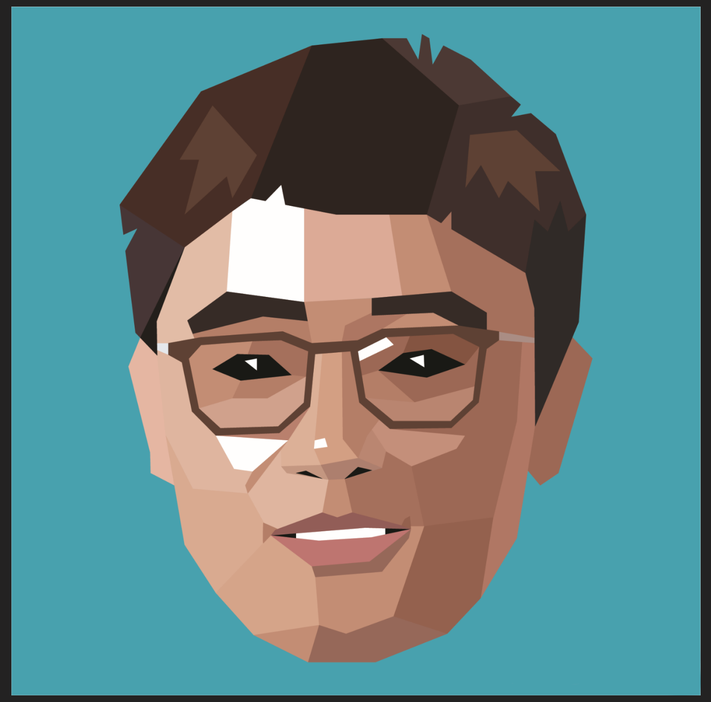
Thor’s hit stun animation

Combat Reaction

- The inertia of the jiggle bone derives fro
  m
  external forces
- Initial velocity is determined on hit direction and hit magnitude
- Jiggle reaction happens in the specific part being attack

Jiggle Squad

Lacognata Nic
Programmer



Tenghao Wang
Senior Technical Artist

Lopez Adrian
Senior Combat Designer

Thank You!

Thank You!

Character Tech Art

Axel Stanley-Grossman
Stephen Miranda
Adrian Rodriguez
Rob Baer
Marisa Kaupert
Snyder Aaron
Johnson Brooke

Animation

Kim Nguyen
Patrick Scanlan
Grace Pan
Fabian Johnston
Erica Pinto

Programming

Nicolas LaCognata
Mathew Hendry
Jeff Miller
Dan Lowe
Koray Hagen
Peter Malnai
James Sweeney

Design

Adrian Lopez
Hendry Lee
Denny Yeh

Questions

Thank You!

Tenghao.wang@sony.com
Tenghaowang720@gmail.com
https://www.linkedin.com/in/tenghaowang/

Santa Monica Studio

STUDIOS