Skeletal Animation

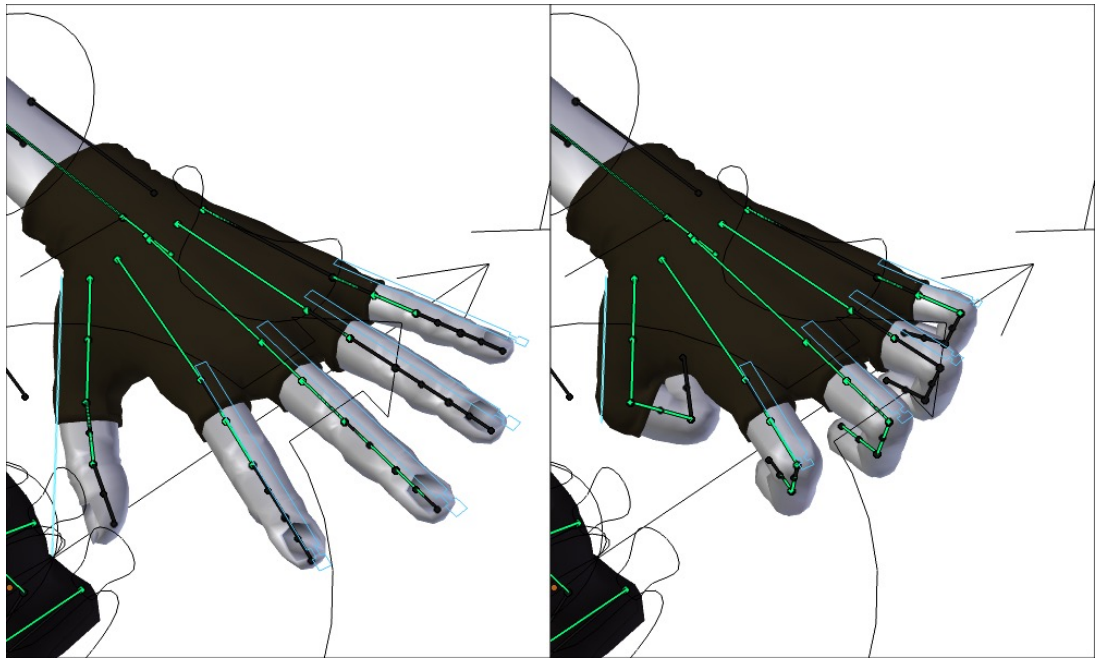
Outline

- Introduction
  - Main Components
- Animation
  - Interpolation
- Skinning
  - How to bind skin vertices to bones?
  - How to compute vertex positions?
- References

Introduction

Introduction

- Animate a 3D model .
- Standard way to animate characters or mechanical objects.

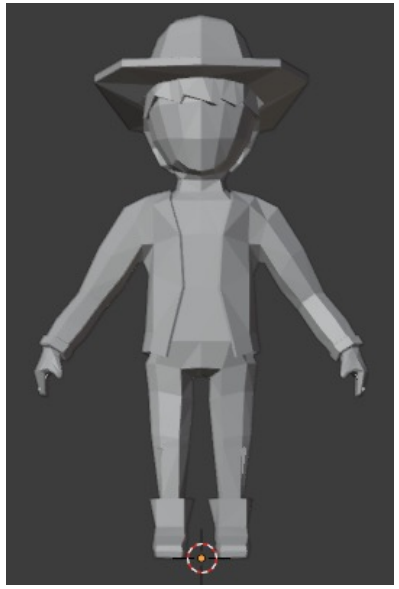


Introduction

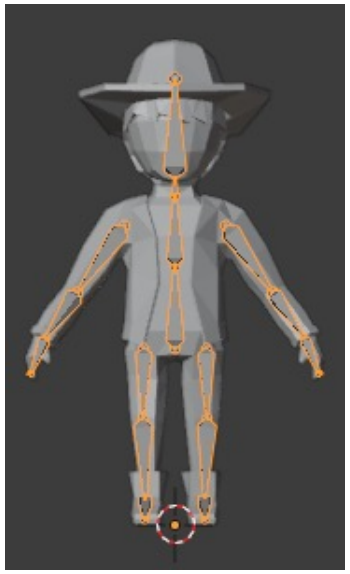
https://learnopengl.com/Guest-Articles/2020/Skeletal-Animation

Introduction

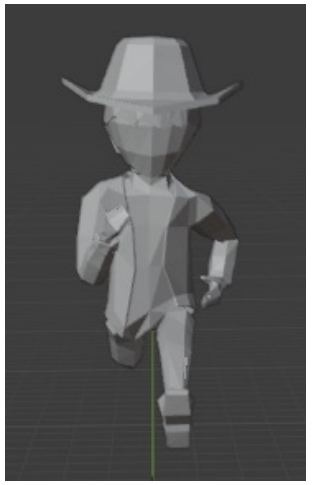
- Main components
  - **Skin** (Mesh)



  - **Bones** (Skeleton)



  - **Keyframes**



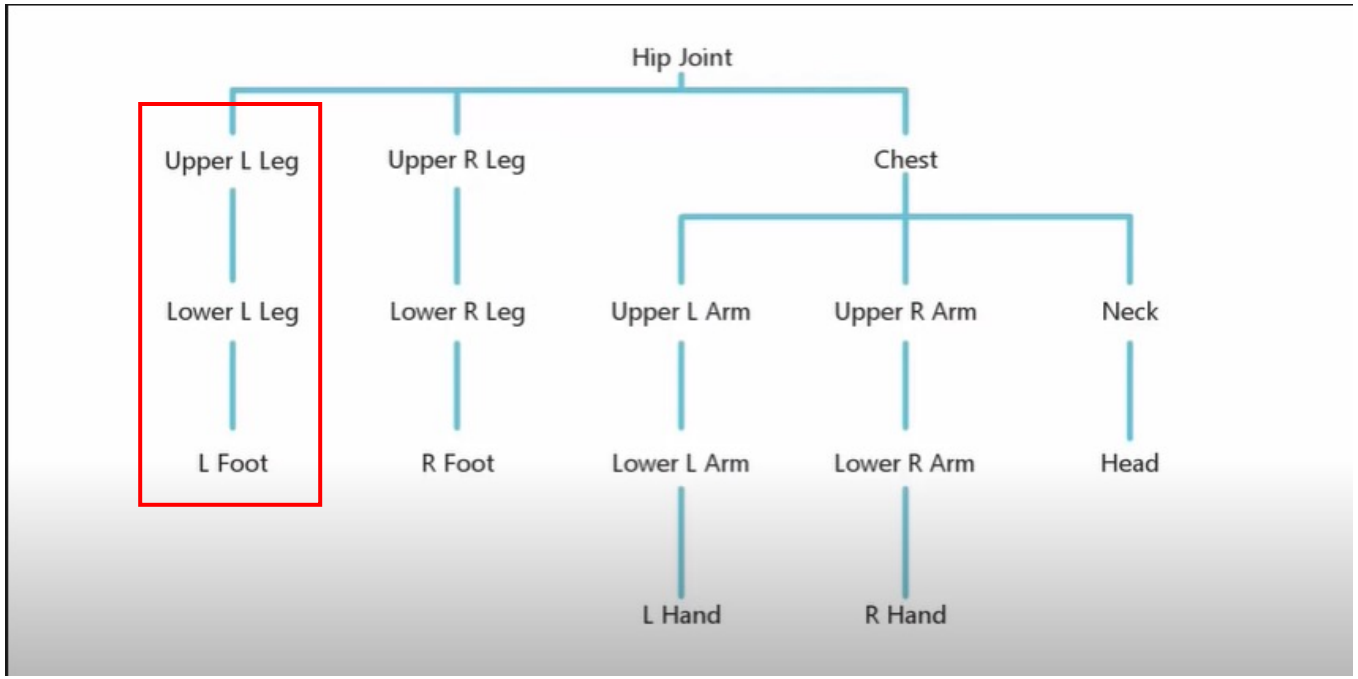
Pose1

Pose2

https://learnopengl.com/Guest-Articles/2020/Skeletal-Animation

Introduction

- The skeleton follows the hierarchy



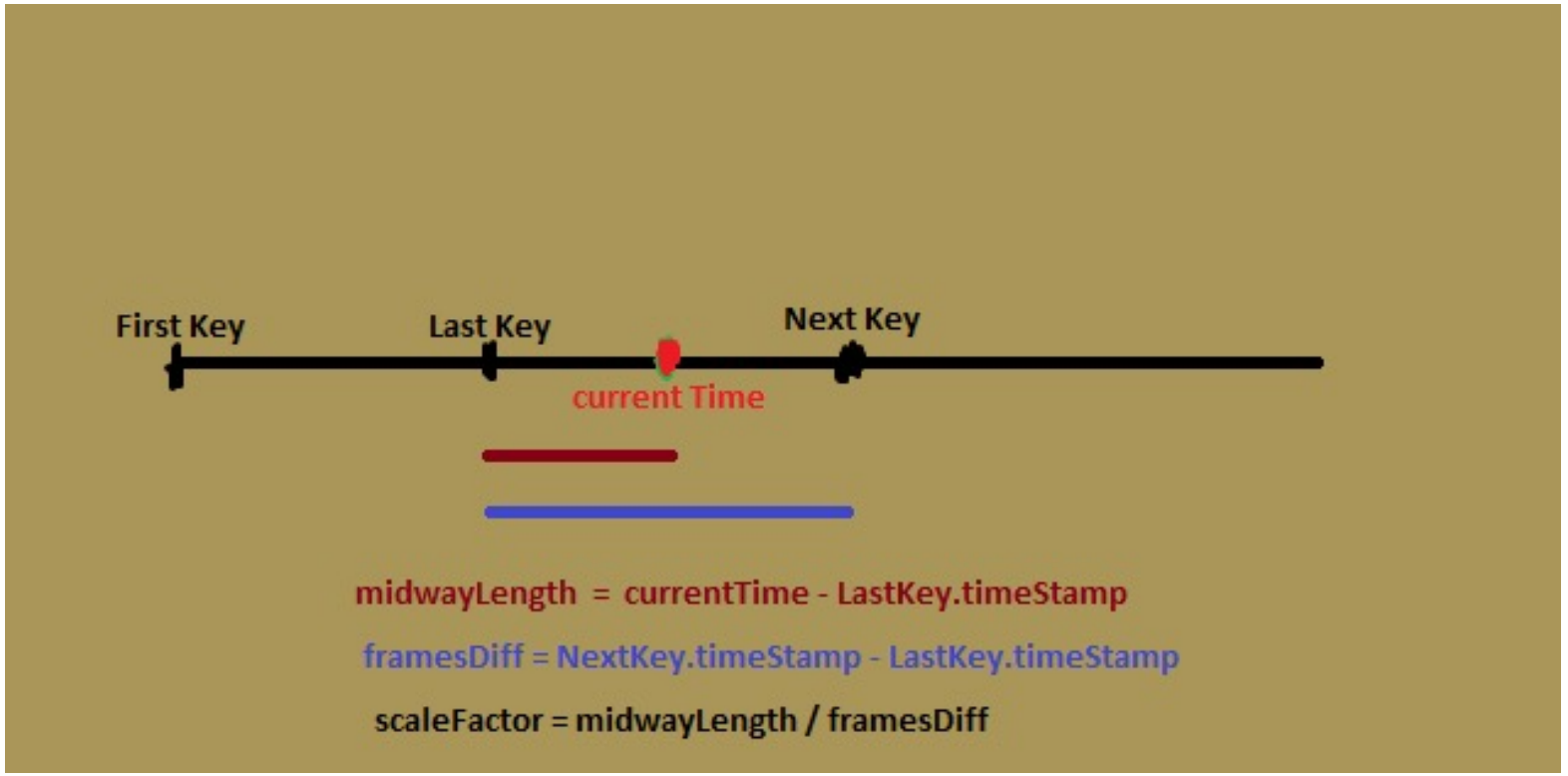
https://learnopengl.com/Guest-Articles/2020/Skeletal-Animation

Animation

Animation

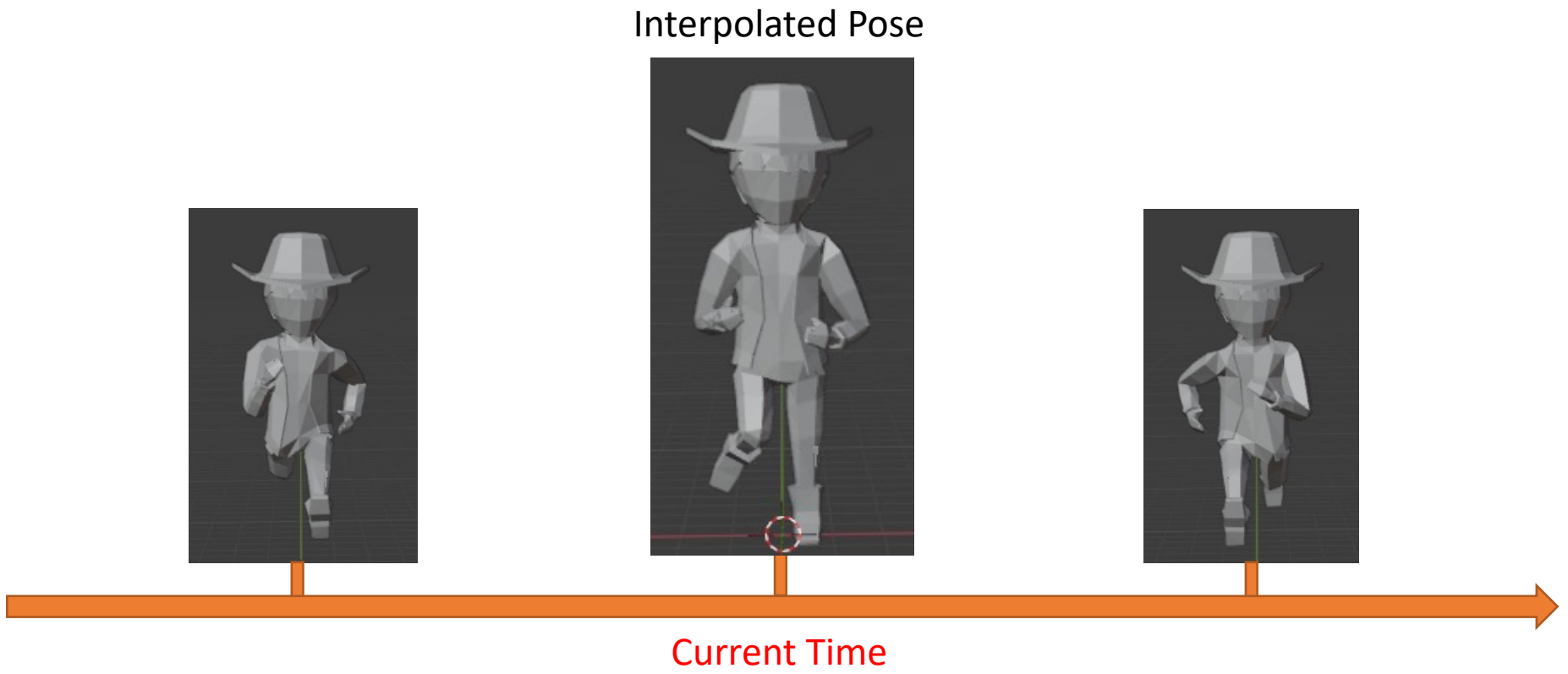
- Keyframes
- Determine the current pose between any two keyframes.
  - **Interpolation**

Animation



https://learnopengl.com/Guest-Articles/2020/Skeletal-Animation

Animation

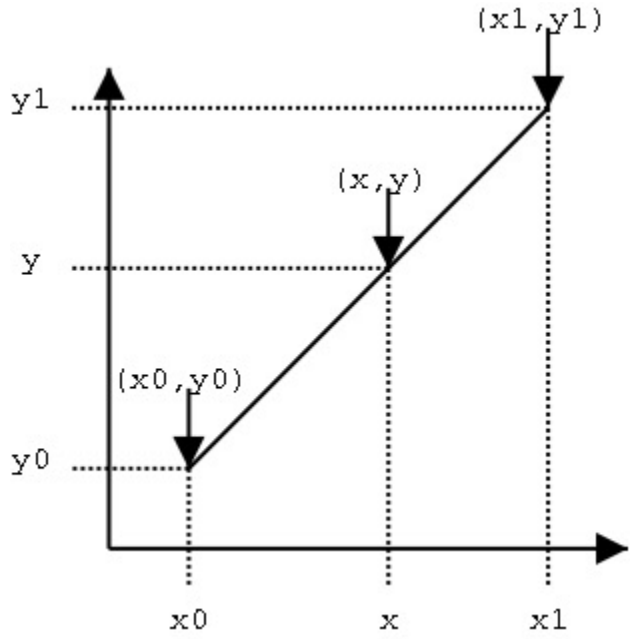


Animation

- How to get the interpolated pose?
  - Interpolating Positions & Scaling
    - Linear Interpolation
  - Interpolating **Rotations**
    - **Spherical Linear Interpolation (Slerp)**

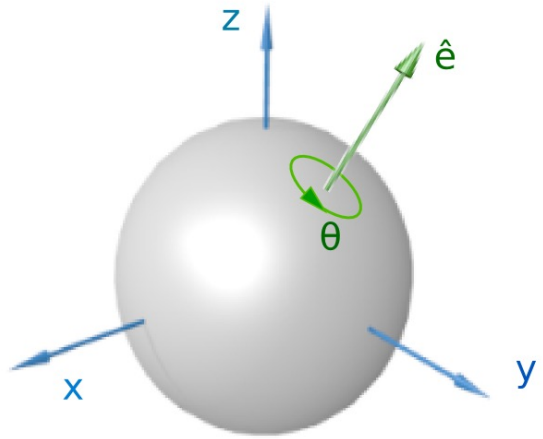
Animation

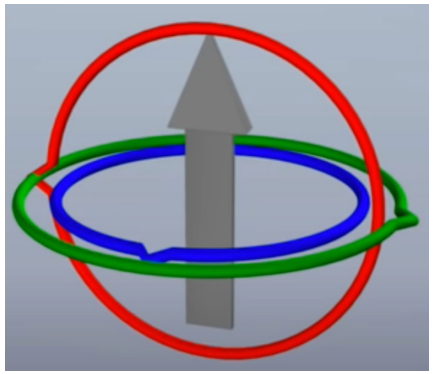
- Linear Interpolation



Animation

- Spherical Linear Interpolation (Slerp)
  - Quaternion
    - a + bi + cj + dk
    - representing spatial orientations and rotations of elements in 3-d space
    - Prevent from **Gimbal lock** problem of Euler Rotation
    - Good for rotation interpolation in 3-d space



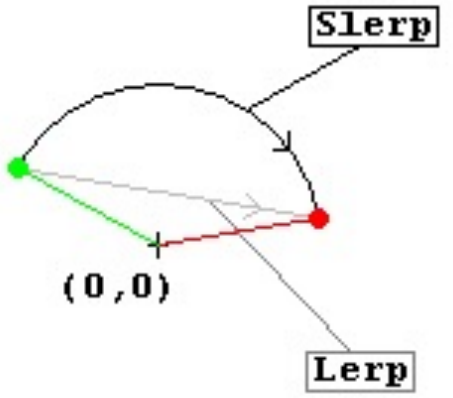


Gimbal Lock

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

Animation

- Spherical Linear Interpolation (Slerp)



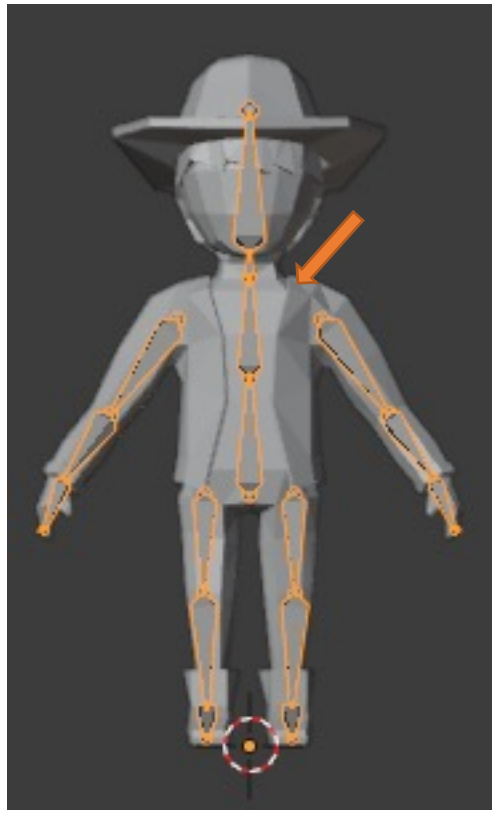

```
glm::quat finalRotation = glm::slerp(m_Rotations[p0Index].orientation,
    m_Rotations[p1Index].orientation, scaleFactor);
```

https://answers.unity.com/questions/389713/detaliled-explanation-about-given-vector3slerp-exa.html

Skinning

Skinning

- How to bind skin vertices to bones?
  - Assign a weight for each vertex for each bone.
  - Limit the number of bones that influences a vertex.



Skinning

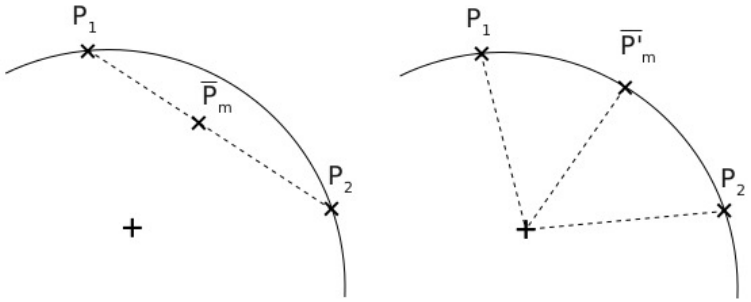
- How to compute vertex positions?
  - Linear blending skinning

```
vec3 skinned =
      BoneWeights.x * ( bones[ BoneIndices.x ] * Position)
    + BoneWeights.y * ( bones[ BoneIndices.y ] * Position)
    + BoneWeights.z * ( bones[ BoneIndices.z ] * Position)
    + BoneWeights.w * ( bones[ BoneIndices.w ] * Position);
gl Position = mvp * vec4(skinned, 1.0);
```

http://graphics.cs.cmu.edu/courses/15-466-f17/notes/skinning.html

Skinning

- How to compute vertex positions?
  - Dual quaternion skinning



$$\dot{\mathbf{q}} = \frac{\sum_{i=1}^{n} w_i \dot{\mathbf{q}}_i}{\| \sum_{i=1}^{n} w_i \dot{\mathbf{q}}_i \|}$$

linear blending skinning

Dual Quaternion Skinning

https://www.cs.utah.edu/~ladislav/kavan08geometric/kavan08geometric.html

References

References

- https://learnopengl.com/Guest-Articles/2020/Skeletal-Animation
- https://www.youtube.com/watch?v=f3Cr8Yx3GGA&t=5s
- https://en.wikipedia.org/wiki/Skeletal_animation
- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-837-computer-graphics-fall-2012/lecture-notes/MIT6_837F12_Lec06.pdf
- Quaternion
  - https://www.youtube.com/watch?v=d4EgbgTm0Bg
- Quaternion & 3D rotation
  - https://www.youtube.com/watch?v=zjMuIxRvygQ
- Gimbal lock
  - https://www.youtube.com/watch?v=zc8b2Jo7mno