

110550047 Animation HW2

Introduction

This project is to practice Kinematics system to standard people bones and import a series of motions to simulate walking and shooting.

Concerning topics includes 3D-Rotations, Kinematics and Keyframing.

Fundamentals

Coordinates

Global coordinates refer to a fixed, absolute coordinate system that is consistent across an entire scene or project. It serves as a reference point for positioning objects relative to the world or scene origin.

Local coordinates, on the other hand, are relative to individual objects or parenting system within a scene. They define an object's position, rotation, and scale relative to its own local coordinate system, which may be different from the global coordinate system.

Rotations

Homogeneous Coordinates

$$\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right) = [x, y, z, w]$$

$$(x, y, z) = [x, y, z, 1]$$

Euler Angle

$$E = (x, y, z)$$

Quaternions

$$Q = (s, x, y, z) \text{ or } (s, v)$$

Kinematics

Kinematics is the branch of physics that studies the motion of objects without considering the forces that cause that motion. It focuses on the position, velocity, and acceleration of objects, as well as their relationships and patterns of motion.

Keyframing

Keyframing allows animators to control various properties such as position, scale, rotation, opacity, and more, over time. By adjusting the values and implementing interpolation of these properties at different keyframes, animators can create lifelike movements, transitions, and visual effects.

Implementation

Forward Kinematics

```
// Initialize
// BFS travel
while(!bones.isEmpty())
{
    bone = bones.pop();
    // Apply Kinematics
    Vi = vi*Length;
    R = RiRj...;
    Ti = R_{i+1} * V_{i+1} + T_{i-1};

    // Breadth-First
    bones.add(bone.siblings);
}
```

Acclaim file

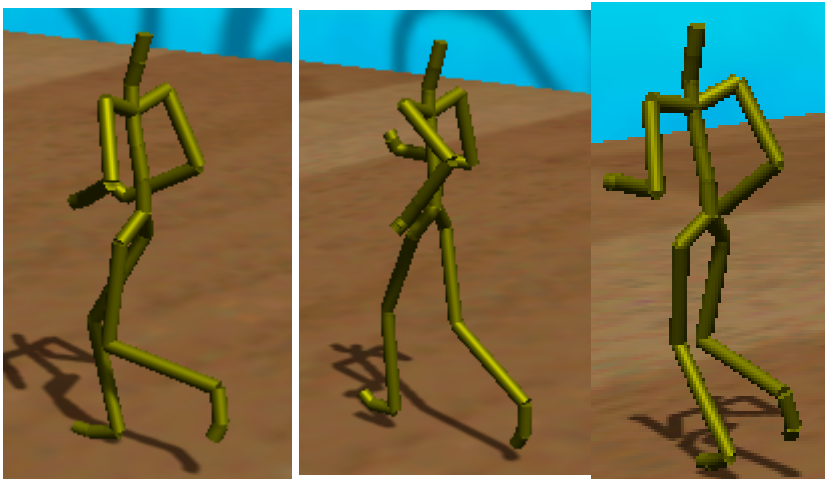
Contains skeletal animation data, including bone rotations, translations, and other parameters, captured from real-world motion. Acclaim files are widely used in fields such as animation, gaming, and biomechanics, serving as a standard format for storing and exchanging motion capture data between different software and hardware systems.

Keyframing

```
double scaler = frame_old / frame_new;
foreach(bone in bones)
{
    leftframe = i * scaler;
    right = i * scaler;
    ratio = bone in [leftframe, rightframe];
    // Lerp Interpolation
    T = (1-ratio)*T[leftframe] + ratio * T[rightframe];
    // Slerp Interpolation
    R = R[leftframe].slerp(R[rightframe], ratio);
}
```

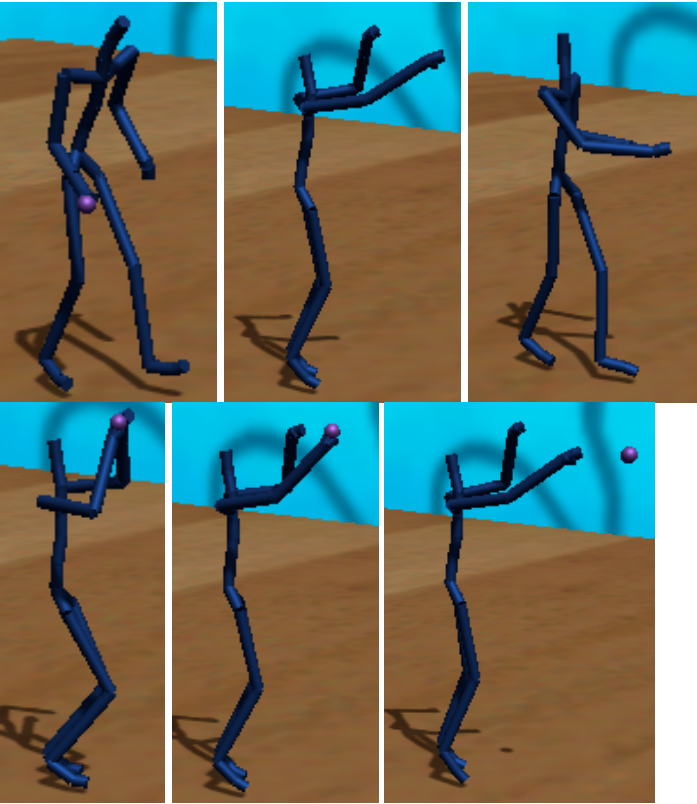
Result and Discussion

Forward Kinematics

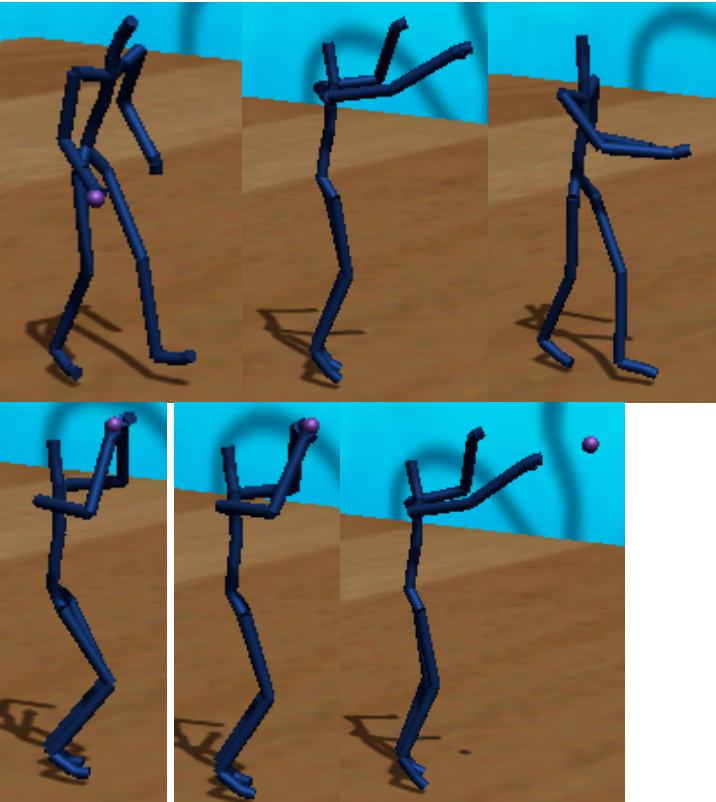


Time Warping

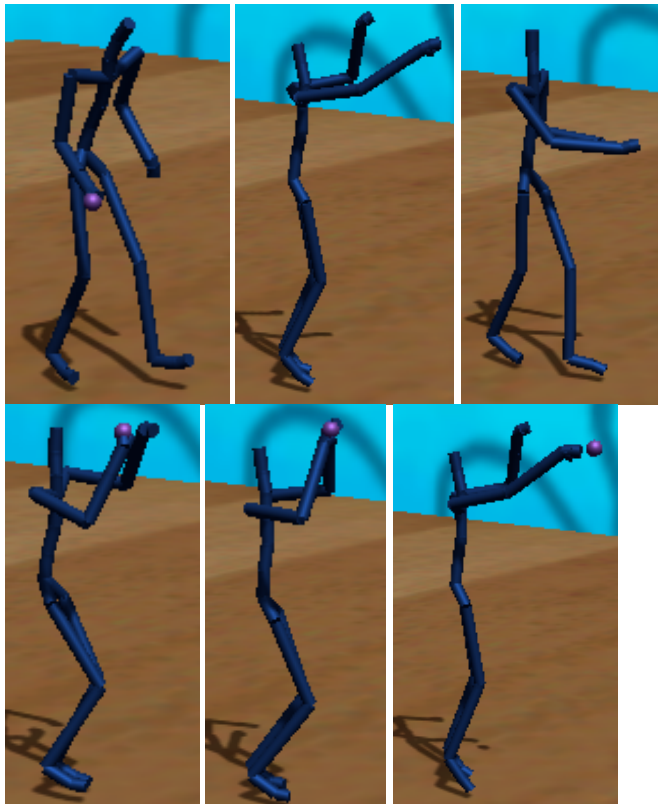
- 328 Frames



- 164 Frames



- 492 Frames



Rotation System

- `util::rotateDegreeZYX`, commonly expressed by rotating X first, then rotating Y , end up rotating Z, convert into $Q = ZYX$ ◦
- The [official document](#) shows `toRotationMatrix().eulerAngles(0, 1, 2)`, which convert into $Q.M = XYZ$.
- `util::rotateDegree` and `toRotationMatrix().eulerAngles` order need to be same ($Q=Q.M=XYZ$ or ZYX), which can be implemented with
 1. `util::rotateDegreeZYX` with `toRotationMatrix().eulerAngles(2, 1, 0)`
 2. `util::rotateDegreeXYZ` with `toRotationMatrix().eulerAngles(0, 1, 2)`
- It is more handy if we use Quaternions to represent any of motions. It would be a bit confusing when converting between Quaternions and Euler Angles. In addition, we need to construct a criterion to specify rotation matrix.

Conclusion

In conclusion, the introduction of forward kinematics and keyframing techniques in animation and motion graphics opens up new possibilities for creating dynamic and realistic animations. Forward kinematics allows animators to create complex movements and articulation of joints, providing a realistic and natural motion for characters or

objects. Keyframing, on the other hand, provides precise control over the timing and spacing of animations, allowing for smooth transitions, dynamic movements, and visually appealing effects.

The use of global and local coordinates further enhances the accuracy and flexibility of animations, enabling animators to position and transform objects in 3D space with precision. Global coordinates provide a consistent reference point for positioning objects in relation to the world or scene origin, while local coordinates allow for object-specific transformations relative to their own coordinate system. Understanding the distinction between global and local coordinates is crucial for maintaining proper spatial relationships in a scene.

Furthermore, the use of Acclaim files as a standard format for motion capture data facilitates the integration of realistic motion data into animations. This allows animators to capture real-world motion and apply it to virtual characters or objects, resulting in more authentic and lifelike animations.