Animation Techniques in Solar Explorer: A Deep Dive

Core Animation Mechanisms

1. CSS Transform and Perspective

The core of the 3D animation is managed through CSS transforms and perspective in the `SolarExplorer.module.scss` file:

```scss
.viewport {
  position: absolute;
  transform: rotatex($sceneAxisTilt);
  perspective: $globalPerspective;
  transform-style: preserve-3d;
  transition: all 1.5s cubic-bezier(0.33, 0, 0, 1);
}
```

Key Parameters:

- `rotatex()`: Tilts the entire scene
- `perspective`: Creates depth effect
- `transform-style: preserve-3d`: Maintains 3D positioning of child elements

2. Planet Transformation Logic

In `SolarSystem.tsx`, planet positioning is dynamically calculated:

typescript

```typescript
const getPlanetStyles = (planet: PlanetType, index: number) => {
  const planetIndex = planetsOrder.findIndex((p) => p === planet);
  const selectedIndex = planetsOrder.findIndex((p) => p === selectedPlanet);
  const distance = index - selectedIndex;

  // Dynamic 3D positioning
  const translateZ = -2800 * distance;
  const translateY = distance * 300;
  const scaleValue = distance === 0 ? 1 : Math.max(0.2, 1 - Math.abs(distance) * 
  const opacity = Math.max(0, 1.5 - Math.abs(distance) * 0.6);

  return {
    transform: `translateZ(${translateZ}px) translateY(${translateY}px) rotatex(4
    opacity,
    transition: "transform 2.5s cubic-bezier(0.19, 1, 0.22, 1), opacity 2s cubic-
  };
};
```

10 Animation Tasks for Sound System Catalogue

1. Change Scene Tilt

Location: `SolarExplorer.module.scss`

Variable: `$sceneAxisTilt`

scss

```scss
$sceneAxisTilt: -35deg; // Adjust this value to change overall scene angle
```

- Decrease for flatter view

- Increase for more dramatic angle

2. Modify Depth Perspective

Location: `SolarExplorer.module.scss`
Variable: `$globalPerspective`

```scss
$globalPerspective: 2500px; // Adjust to change depth perception
```

- Smaller values: More compressed 3D effect
- Larger values: More spread-out 3D space

3. Customize Item Spacing

Location: `SolarSystem.tsx`
Modify Calculation:

```typescript
const translateZ = -2800 * distance; // Adjust multiplier for spacing
const translateY = distance * 300;   // Adjust for vertical spacing
```

- Change multipliers to adjust item positioning
- Experiment with different values for unique layouts

4. Create Dynamic Scaling

Location: `SolarSystem.tsx`
Scaling Logic:

typescript

```
const scaleValue = distance === 0
  ? 1
  : Math.max(0.2, 1 - Math.abs(distance) * 0.35);
```

- Adjust multiplier (`0.35`) to control scaling rate
- Change `0.2` to set minimum scale

5. Control Opacity Transition

Location: `SolarSystem.tsx`

Opacity Calculation:

typescript

```
const opacity = Math.max(0, 1.5 - Math.abs(distance) * 0.6);
```

- Modify `1.5` to adjust starting opacity
- Change `0.6` to control fade rate

6. Customize Transition Timing

Location: `SolarExplorer.module.scss`

Transition Parameters:

scss

```
transition: transform 2.8s 0.3s cubic-bezier(0.33, 0, 0, 1);
```

- Adjust duration (`2.8s`)

- Modify delay (`0.3s`)
- Change cubic-bezier for different easing

7. Implement Rotation Animation

Location: `SolarExplorer.module.scss`

Keyframe Animation:

```scss
@keyframes item-rotate {
  0% { background-position: 0 0; }
  100% { background-position: 1140px 0; }
}

.item {
  animation: item-rotate 60s infinite linear;
}
```

- Adjust duration (`60s`)
- Change `linear` to other timing functions

8. Create Hover Effects

Location: `SolarExplorer.module.scss`

Hover Transformation:

scss

```scss
.item {
  transition: transform 0.3s ease;

  &:hover {
    transform: scale(1.05) translateZ(50px);
    box-shadow: 0 10px 20px rgba(0,0,0,0.2);
  }
}
```

- Adjust scale factor
- Modify shadow and translation

9. Implement Responsive Scaling

Location: `tailwind.config.js`

Responsive Breakpoints:

javascript

```javascript
module.exports = {
  theme: {
    screens: {
      'sm': '640px',
      'md': '768px',
      'lg': '1024px',
      'xl': '1280px',
    },
    extend: {
      scale: {
        '95': '0.95',
        '105': '1.05',
      }
    }
  }
}
```

- Add media queries
- Create responsive scale transformations

10. Background and Overlay Animations

Location: `SolarExplorer.module.scss`

Background Effect:

```scss
.background-overlay {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: linear-gradient(45deg, #dark-color, #light-color);
  animation: gradient-shift 10s ease infinite;
}

@keyframes gradient-shift {
  0% { background-position: 0% 50%; }
  50% { background-position: 100% 50%; }
  100% { background-position: 0% 50%; }
}
```

- Modify gradient colors
- Adjust animation duration and timing

Animation Performance Tips

- Use `transform` and `opacity` for best performance
- Leverage CSS `will-change` for complex animations
- Use `requestAnimationFrame` for JavaScript animations
- Minimize repaints and reflows

Debugging Techniques

1. Use browser developer tools

2. Check performance tab
3. Use CSS `outline` for visualization
4. Implement gradual changes

Would you like me to elaborate on any of these animation techniques or discuss how to specifically adapt them for sound system equipment?