I have been working as a software engineer with Adobe in computer graphics for the past few years. Specifically, I have been a part of a team that develops software for photorealistic 3D rendering. It is a virtual staging software that allows you to import 3D assets, then arrange and stage them to build a scene. Users can add lights to further customize the appearance of the scene. Commonly, virtual staging software uses a camera-based system to allow users to set up shots for rendering. The rendering techniques used are mixed - both real-time and ray-tracing rendering is supported. I deal with both discrete and continuous events in the application from time to time.

Some of those events are described below:

**Discrete Events:**

1. **Rasterization**: Rasterization is the process of converting geometric primitives such as triangles into pixel data that can be displayed on screen. The continuous geometric information is transformed into discrete pixels.

Impact: The developer needs to ensure the appropriate attributes and the algorithm.

1. **User input events**: User interactions such as mouse clicks, keyboard input or touchscreen gestures.

Impact: The developer needs to implement event handling to respond to these events. This might include updating the scene or camera based on user actions.

1. **Mesh Subdivision**: Mesh subdivision is an application of geometry processing. It involves increasing the detail of the mesh through addition of vertices and faces.

Impact: while implementing mesh subdivision, the developer needs to maintain mesh smoothness and avoid artifacts.

1. **Curve and surface fitting**: This is the process of fitting curves or surfaces to a set of points.

Impact: Developers need to address data noise, choice of fitting functions in curve fitting.

**Continuous Events:**

1. **Animation Playback**: Animation playback is a continuous event. It involves continuously updating and rendering frames to create the illusion of motion.

Impact: The developer needs to interpolate between different states of 3D over time and manage animation sequences.

1. **Camera Movement**: Continuous movement of the virtual camera to capture the 3D scene.

Impact: Developers need to implement camera control systems

and user inputs for camera movement.

1. **Physics Simulation**: Simulation of physical behavior – object motion, particle systems, collisions.

Impact: Developers must implement and optimize physics engines, considering factors like stability and accuracy.

1. **Lighting Changes**: Changes to lighting conditions, such as the movement or adjustment of light sources.

Impact: Developers need to create algorithms that update lighting parameters in real-time, ensuring realistic rendering and adapting to changes in the environment.

Other Impact:

1. Complexity Management: Both discrete and continuous events introduce complexity into the development process.
2. Performance Optimization: Developers must optimize code for real-time rendering and utilization of hardware acceleration.
3. User Experience: Both types of events contribute to the overall user experience. Discrete events, such as loading models or textures, can impact initial load times, while continuous events, like smooth animation and dynamic lighting, contribute to an immersive and visually appealing user experience.