### **Unit 6 Learning Journal**

This week, I spent a considerable amount of time working on an animation project using Three.js. The primary focus was to create a simulation of Earth and Moon rotation, along with a moon orbit around the Earth, while accounting for lighting effects that resemble day, night, and even an eclipse. This task involved using sphere geometries, textures, lighting, and rotation functions, which was an excellent exercise to solidify my understanding of 3D graphics and animations in JavaScript.

One of the most interesting aspects of this task was learning how to apply textures to 3D models, making them look realistic. Applying textures from external resources, such as images of the Earth and the Moon, helped me appreciate how simple 2D images can enhance the realism of 3D models. Furthermore, creating rotations and orbits to mimic real-world phenomena such as Earth’s day/night cycle and lunar eclipses was challenging but rewarding.

### **Surprises and Challenges**

One of the biggest surprises for me was the amount of effort required to fine-tune the appearance and movement of 3D objects. Although the theoretical concepts behind rotation and light sources seem simple, getting everything to work correctly was more complex than anticipated. For instance, the positioning of the light source was tricky because I wanted to simulate how the sun only lights up one side of the Earth, creating a realistic day/night effect. Fine-tuning the lighting settings required a lot of trial and error, which I didn’t expect.

Another challenge I encountered was dealing with the texture wrapping of the spheres. At first, the textures didn't align properly on the Earth and Moon, which created distorted visuals. After researching, I discovered that tweaking the UV mapping of the textures could solve this issue. This was a new concept for me, and it added to my understanding of how graphics rendering works in Three.js.

To overcome these challenges, I relied heavily on documentation and tutorials. Reading through the official Three.js documentation and watching videos on YouTube helped me understand how to apply textures correctly and how to position lights for realistic effects. The active Three.js community was also a valuable resource, as I found several answers to common problems on forums like Stack Overflow.

### **How Computer Animation Techniques Have Revolutionized Cinema Arts**

Computer animation techniques have dramatically transformed the movies and cinema arts. The advancements in computer graphics, rendering techniques, and animation algorithms have enabled filmmakers to create immersive, visually stunning experiences that were previously unimaginable. One of the most significant innovations in this field is the development of CGI (Computer-Generated Imagery), which allows animators and filmmakers to generate lifelike visuals and intricate environments.

Take, for example, the **Ray Tracing** technique. According to Glassner (1989), ray tracing simulates the way light interacts with surfaces, producing highly realistic reflections, shadows, and textures. This technique is widely used in animation and special effects to create lifelike environments, characters, and objects that are indistinguishable from reality. A great example of this is Pixar’s animated films, where the combination of physics-based animations and ray tracing brings depth and realism to scenes.

In addition to CGI, **motion capture** has become a game-changer for animators and filmmakers. Motion capture involves recording the movements of real actors and using that data to animate 3D characters. This technique is widely used in films like *Avatar* (2009), where the actors' performances are mapped onto digital characters, creating seamless transitions between live action and animation. According to Chang (2009), the integration of motion capture with CGI has allowed for the creation of highly expressive and emotive characters, which was previously difficult to achieve through traditional animation techniques.

Another noteworthy advancement is **procedural animation**, which allows the automatic generation of animations using algorithms. Procedural animation is commonly used for large crowds, natural elements (such as water or fire), and complex phenomena that would be impossible to animate manually. Films like *The Lord of the Rings* trilogy use procedural animation for scenes that involve thousands of characters, making it possible to create realistic crowd behaviors without animating each individual character by hand.

Moreover, **physically-based rendering** (PBR) has improved the way materials interact with light, providing more accurate representations of surfaces such as metal, glass, and skin. This has opened the door for highly realistic visual effects in movies, giving filmmakers the ability to create virtual worlds that feel grounded in reality.

In conclusion, the introduction of advanced computer animation techniques has revolutionized the cinema industry, enabling filmmakers to create more immersive and visually compelling stories. Techniques like CGI, motion capture, ray tracing, and procedural animation have significantly improved the quality of animated films and special effects, making it easier to create detailed, lifelike worlds that captivate audiences.

### **References**

[Glassner, A. S. (1989). *An introduction to ray tracing*. Academic Press.](https://shop.elsevier.com/books/an-introduction-to-ray-tracing/glassner/978-0-08-049905-5)

This entry reflects on both the technical challenges faced this week and the broader impact of computer animation techniques in film. It serves as a documentation of the progress made and the continuous learning process in mastering animation technologies.