5. **Input and Event Handling**
   - Mapping hardware inputs (keyboard, mouse, controller) to in-game actions.
   - Designing custom input systems for flexibility and extensibility.
6. **Audio Systems**
   - Basics of sound processing and integration into games.
   - 2D and 3D audio spatialization for immersive soundscapes.
7. **Tools and Asset Management**
   - Building asset pipelines for models, textures, and animations.
   - Optimization strategies for loading, managing, and utilizing assets efficiently.
8. **Networking and Multiplayer**
   - Architectures for real-time multiplayer games (client-server and peer-to-peer).
   - Synchronizing game state across clients and managing latency.
9. **Scripting Systems**
   - Embedding scripting languages (e.g., Lua, Python) to define game logic dynamically.
   - Benefits of scripting for rapid prototyping and customization.
10. **Parallelism and Optimization**
    - Techniques for parallel processing and multithreading.
    - Profiling and debugging tools for identifying performance bottlenecks.

Learning Outcomes

- Develop a solid understanding of the key components of a game engine.
- Gain hands-on experience designing and implementing modular systems.
- Evaluate and compare game engines for different genres and use cases.
- Learn optimization techniques to ensure smooth and responsive gameplay.