Game Systems List with Potential Class Names (C# & ECS):

I. Core Engine Systems:

1. **Rendering System (2D):**
    - `SpriteComponent`, `AnimationComponent`, `SpriteLoadingSystem`, `SpriteRenderingSystem`
    - `Texture`, `TextureLoadingSystem`
    - `Renderer`, `RenderCommand`
    - `CameraComponent`, `CameraSystem`
    - `Animation`, `AnimationState`, `AnimationSystem`
2. **Input System:**
    - `InputStateComponent`, `KeyboardInputSystem`, `MouseInputSystem`, `GamepadInputSystem`
    - `InputAction`, `InputMappingSystem`
3. **Game Loop:**
    - `Game` (main game class), `GameTime`
4. **Resource Management:**
    - `Resource`, `ResourceCache`, `ResourceLoader`
    - `DevModeConfig`
    - `ResourceLoadingSystem`
5. **World/Level Management:**
    - `LevelData`, `TilemapComponent`, `TileData`
    - `LevelLoadingSystem`, `TilemapRenderingSystem`, `AreaTransitionSystem`
    - `TransformComponent`
6. **Player Character System:**
    - `PlayerComponent`, `MovementComponent`, `StaminaComponent`, `HealthComponent`, `InventoryComponent`, `StatsComponent`
    - `PlayerMovementSystem`, `PlayerCollisionSystem`, `PlayerStaminaSystem`, `PlayerHealthSystem`, `PlayerInventorySystem`, `PlayerStatsSystem`
7. **Combat System:**
    - **Combat Management:**
        - i. `CombatSystem`
        - ii. `TurnManager`
        - iii. `InitiativeSystem`
    - **Entity Components:**
        - i. `CombatantComponent`
        - ii. `ActionPointsComponent`

   iii. `TurnOrderComponent`
   iv. `IntentComponent`
   v. `AbilityComponent`
   vi. `TargetComponent`
   vii. `EffectComponent`

- **Action Handling:**
   i. `Action (Base class or Interface)`
   ii. `ActionProcessor`
   iii. `AttackProcessor`
   iv. `MoveProcessor`
   v. `AbilityProcessor`
   vi. `ItemProcessor`
   vii. `Ability System:`
   viii. `AbilityData`
   ix. `AbilityActivationSystem`
- **Targeting System:**
   i. `Targeting`
- **UI Presentation (Related Systems):**
   i. `CombatUISystem`
   ii. `ActionSelectionUI`

8. **Crafting/Forging System:**
   - `SpiritData`, `CraftingRecipe`, `CraftingResult`
   - `CraftingUISystem`, `ForgingLogicSystem`
9. **User Interface (UI) System:**
   - `UIComponent`, `TextComponent`, `ImageComponent`, `ButtonComponent`
   - `MenuSystem`, `HUDSystem`, `InventoryUISystem`, `CraftingUISystem`, `DialogueSystem`, `NotificationSystem`
10. **Dialogue and Narrative System:**
    - `DialogueData`, `DialogueNode`, `NPCScript`
    - `DialogueUISystem`, `NPCScriptingSystem`
11. **Save/Load System:**
    - `SaveData`, `SaveLoadSystem`
12. **Audio System:**
    - `SoundEffect`, `MusicTrack`
    - `SoundEffectSystem`, `MusicSystem`
13. **Networking System:**
    - **Connection Management:** `NetworkManager`, `Client`, `Server`, `Connection`
    - **Network Components:** `NetworkIdentityComponent`, `NetworkTransformComponent`, `NetworkStateComponent`

- **Serialization/Deserialization:** `NetworkSerializer`, `NetworkReader`, `NetworkWriter`
- **Message Handling:** `NetworkMessage`, `MessageHandler`
- **Transport Layer:** `NetworkTransport`
- **Synchronization Systems:** `NetworkTransformSystem`, `NetworkStateSynchronizationSystem`
- **Event Handling:** `NetworkEvent`, `NetworkEventListener`

14. **Configuration System:**
    - `GameConfig` (general game settings, e.g., initial screen size)
    - `GraphicsConfig` (resolution, fullscreen, etc.)
    - `AudioConfig` (volume levels)
    - `InputConfig` (keybindings)
    - `NetworkConfig` (server address, port)
    - `ConfigurationManager` (loads, saves, and manages configuration data)
15. **In-Game Settings System:**
    - `SettingsData` (stores the current in-game settings)
    - `SettingsManager` (loads, saves, and applies in-game settings)
    - `SettingsMenu` (UI for the in-game settings)
    - `GraphicsSettingsSystem` (applies graphics settings)
    - `AudioSettingsSystem` (applies audio settings)
    - `InputSettingsSystem` (applies input settings)
16. **Optional/Advanced Systems:**
    - `ParticleComponent`, `ParticleEmitterComponent`
    - `ParticleSystem`
17. **Data-Driven Design and Binary Packaging System:**
    - `DataLoader`, `BinaryPacker`
    - Configuration classes for loading modes

II. Packages:

1. Nuke.Common
2. SDL3-CS
3. Newtonsoft.Json
4. ENet-CSharp