FX Infrastructure

FX-related Things to build for the game: tools, pipeline, and runtime systems.

- **Weather system:** Precipitation, diegetic fog and debris, integration with lighting and atmospherics. Could be an extension of Sky Creator, or a separate system.
  - Weather as a gameplay element?
  - Environment puzzles?
  - Unified force system which can affect Niagara systems, possibly foliage, Chaos systems.
  - Investigate Pivot Painter.
  - ATLAS-567: [FX] Weather, wind, precipitation, etc. BLOCKED
- ~~**Sand System:** Setups for shifting, blowing sand; volumetrics. (As needed by game)~~
  - ~~Weapons turn sand to glass? (Weapons affect environment in general?)~~
  - DRD-10446: [FX] Complex animation of shifting sand dunes WAIVED
- **Footprints:** Dynamically generated terrain deformation, preferably integrated with movement (as opposed to manually timed AnimNotifies).
  - Persistence?
  - DRD-10445: [FX] Visual and other FX related to footprints; footprint detection and triggers WAIVED
- **Rich VFX Toolset:** Drop-in, artist-oriented solution for spawning and managing “rich” effects: post, decals, Niagara systems, material-based effects, etc. Rich effects are entities combining one or more of these assets and flexible, possibly complex timing.
  - ATLAS-514: [Suite of native gameplay tools for composite FX on Hit Impacts and in Blueprints TO DO
- **Performance Monitoring HUD:** Enhance in-game display of FX-related performance metrics, and budgeted times in general.
  - DRD-10447: [FX] Custom performance monitoring HUD; at-a-glance tools for artists to recognize perf issues WAIVED
- **User-level HLSL Support:** Develop a plugin container for HLSL shaders. These would appear as material functions in the editor, but exist as visible HLSL in the code base.
  - ATLAS-542: [FX] User-level HLSL TO DO
- **FX-related asset validation:** Add specific checks for things like setting Niagara emitters to GPU, ensuring all Niagara systems have near/far fade settings, etc.
  - ATLAS-499: VFX Asset validation TO DO
- ~~**Destructibles:** A flexible, modular, stable, and performant system for Chaos sims. Needs to be easily configurable and usable by level design. One approach could be a library of predefined “Lego” pieces.~~
  - ~~Are destructibles gameplay elements? (Barriers, building blocks)~~
  - ~~Hiding treasure? (encourage exploration)~~
  - DRD-10450: [FX] Destructible gameplay and environmental objects WAIVED
- **Aggressive, Dynamic Runtime FX Optimization:** Add features like removing distant Niagara systems after a timeout and other forms of culling which don’t exist in Unreal; object pooling; caching/uncaching, etc.
- **Aggressive, Build-time FX Optimization:** Pruning disabled, but unused, emitters or Niagara systems (cook keeps these to be enabled at runtime, but this feature isn’t actually supported). Also, add any obscure cases Unreal doesn’t handle)
- **Volumetrics Pipeline:** Support for 3D, volumetric rendering. Possibly based on SVTs, VBDs; integration with Houdini.
- **Procedural VFX Distribution:** Automated system to place environment VFX.
- **Character Damage FX:** See visual indications of damage or other states on Amara, possibly enemies.
- **GPU-resident Particle Emitters (stretch R&D project):** Add a Niagara emitter type which doesn’t need to be updated by the CPU on the render thread. This addresses performance issues with very large numbers of ambient VFX: even if they are simulated on the GPU, the emitter position is updated every frame by the CPU. This can be a considerable cost in some cases.
  See also: GDC Enhancement of Particle Simulation Using Screen Space Techniques in 'The Last of Us Part II'