VUE + PLANTFACTORY

Proposal to Join the Academy Software Foundation

TAC meeting 01/21/26

Presenter: Colin Chargy

(Presentation prepared by Colin Chargy & Daniel Seebacher)

Project Sponsor: Industrial Light & Magic

Speaker introduction

- **Colin Chargy:** former member of the VUE/PlantFactory dev team (joined in 2013)
  - Core developer
  - DevOps engineer
  - Linux specialist
- Initial maintainer & lead developer
- Motivation:
  - Providing long-term sustainability,
  - Growing and transferring knowledge to the community
  - Aligning the applications with industry needs

VUE + PLANTFACTORY

A brief history

- Originally developed by e-on software, founded in Paris in 1997 by Nicholas Phelps
- First version **Vue d'Esprit 1** developed by Nicholas Phelps and released in 1997
- Company growth, early introduction of procedural and node-based workflows around 2004/2005
- Large-scale scattering systems and integration plugins for Maya, 3ds Max, Lightwave, Softimage & C4D introduced in 2005
- Volumetric spectral atmospheres introduced in 2006
- First version of **PlantFactory** released in 2013
- E-on software acquired by Bentley Systems in 2015
- Products EOLed & made available for free in 2024 due to company-wide product line restructuring

Artwork by Daniel Respaud

VUE

VUE is an **artist-friendly environment authoring tool** designed to assemble **large environments** quickly and efficiently with **non-destructive workflows and infinite procedural variation**.

Core Capabilities

- EcoSystems & procedural plants (integrated with PlantFactory)
- Terrains & procedural planets
- Environmentally aware materials
- Procedural ocean
- Art-directable spectral atmosphere system with volumetric clouds
- Extensive export & integration capabilities

➔ Accessible frontend for easy workflow & fast iteration, but everything powered
by nodes under the hood

Artwork by Aron Kamolz

PLANTFACTORY

PlantFactory is a comprehensive node-based **procedural vegetation authoring tool** for creating high-quality plant models for both real-time and offline rendering purposes.

PLANTFACTORY

Core Capabilities

- Node-based workflow with support for manual touch-up, pruning, and drawing
- Infinite procedural variations with automatic LOD mesh creation
- Support for season, maturity, and health
- Extensive Preset & Published Parameters system
- High degree of randomness control & advanced growth logic
- Plant models usable standalone (as baked meshes through export & plugins) or integrated within VUE (as procedural assets)

➔ More technically oriented than other modelers, but much more control & flexibility

Rendered with Redshift in C4D using the PlantFactory plugin for automatic Redshift material creation. Artwork by Daniel Seebacher

PlantFactory preset examples within one *.tpf species file each

Palm tree shape presets

Maturity examples

Oak tree shape presets

Season examples

VUE + PLANTFACTORY

Shared foundations and workflow continuity

- Common code base & shared repository
- Shared procedural wind model (scene-wide wind interaction in VUE)
- Common node system (fractals, math, colors, logic…)
- Shared render engine
- Interactive editing of PlantFactory plants from within VUE

➔ coherent authoring workflow from **asset** to **environment**

VUE + PLANTFACTORY

Where we left off in 2024

- Codebase cleanup and refactoring
- Opening the apps up to modern pipeline integration through
  - Extensive export capabilities
  - New plugins (e.g. Unreal Engine with support for Pivot Painter 2 wind, Omniverse connectors for both apps)
- Implementation of the Hydra USD bridge for rendering (tech preview of Cycles inside VUE 2024)
- Preparation for integration of MaterialX
- Full USD support introduced in 2021
- Rewrite of the linear workflow

VUE + PLANTFACTORY

Known limitations

- Legacy CPU-based rendering
- Materials, nodal and atmosphere systems tied to the legacy (CPU-based) renderer
- The apps predate some modern standards such as OpenColorIO, ACES or DeepEXR support
- Viewport engine needs modernization
- Plugins need to be updated after two years of development hiatus
- User-interface is still under development for the Linux version

VUE + PLANTFACTORY

Long-term vision (not product or roadmap specific)

- Preserve, stabilize and extend the procedural authoring core
- Modernize and modularize legacy components over time
- Improve pipeline interoperability with modern standards (benefitting from other ASWF projects)
- Explore new rendering paths with the community (updating legacy renderer? Migrating to a GPU-based path tracer? Incorporating external rendering support through Hydra USD? etc.)
- Improve plugin user experience by relying on each host app's native UI-system and specificities

VUE + PLANTFACTORY

Community & contributors

- Initial contributors include former e-on software team members (devs, support, PM/PO)
- Further contribution interest from Blender add-on developers, former Maxon and Lightwave developers
- Small, but active user community exists for the current freeware versions
- Interest in using the apps has been expressed by
  - Universities for technical research & LLM training,
  - Small studios with visualization and all-in-one software needs,
  - Gaming studios with needs for plant modelling and skyboxes,
  - The Blender community
  - The Geographic Information System community
  - Studios reevaluating their environment pipelines
  - Technical artists focused on procedural workflows

VUE + PLANTFACTORY

Open-sourcing status

- Codebase is not open-source yet
- Ongoing work to separate VUE/PF from other not-to-be-freed IP in the same codebase.
- Chosen licence: Apache 2.0
- Need help for the ASWF to:
  - Create the governance model
  - Define clear contribution rules
  - Transition everything to open-source mindset

VUE + PLANTFACTORY

Transitioning to open-source

- Codebase size and complexity
  - CI/CD, DevOps pipelines, automatic testing as early and often as possible
  - Phased releases (all existing features like plugins might not end up in the initial release)
- Learning curve
  - Documentation for new contributors
  - Code reduction and improvement for readability
- Transition from proprietary to open governance
  - Follow Open-source good practices (for example Security and compliance from the OpenSSF)
  - ASWF guidance & best practices
  -

Product demos coming up next

Then I'll be happy to answer your questions