VIRTUAL FX PACK | Magical Healing VFX

Lily'z Games

Website: www.lilyzgames.com | Discord Community & Support Server:

You will find the Demo Video within the Discord Server :)

Join the Discord Server Now!

1) Overview

This package contains **7 magical healing VFX variations** designed for real-time gameplay readability (RPG / fantasy / MOBA / action). Each variation is meant to cover common healing scenarios such as **instant heal bursts**, **regen loops**, **aura heals**, and **pickup/zone-style healing**.

Included:

- 7 unique heal effects (variations)

- Materials / instances required for the visuals

- Example setup in the level (walkthrough section below)

Not included:

- Blueprints (none in this pack)

- Sound effects

- Characters / environments shown in preview media (if any)

2) Asset Contents / Folder Layout

- Main/Content/NiagaraSystems/

- Main/Content/Maps/

- Main/Content/Materials/

- Main/Content/Textures

3) Quick Start (fastest way to use)

1. **Drag & drop** NS_HealingEffect into your scene.

2. Position it at the character's feet, chest, or slightly above ground depending on style.

3. Choose how it plays:

   ○ **Auto-play**: start on spawn (good for pickups, ambient regen zones)

   ○ **Triggered**: spawn / activate it when healing occurs (recommended for abilities)

4. To modify its appearance, open NS_HealingEffect and check or uncheck emitters boxes inside the file, you can also maybe "Save As" multiple of your custom healing effects!

4) How to Trigger the Effect (no Blueprints included)

Because there are **no Blueprints** in this pack, you trigger the VFX using your engine's standard approach:

Common triggering methods (pick what fits your game):

- **Spawn at Location** when heal happens (instant heal burst)

- **Attach to Character** (regen / aura loop)

- **Activate/Deactivate** an already placed effect (healing zone or persistent aura)

Suggested gameplay uses:

- **Instant heal ability:** spawn burst variation on cast impact

- **Heal over time (HoT):** attach looping variation for duration

- **Healing pickup:** auto-play when player overlaps pickup

- **Healing zone:** place loop in level + enable when player is inside

5) Customization Guide (most important settings)

The pack is designed so you can quickly create multiple looks (holy/nature/arcane/sci-fi) using the same base effects.

Core customization knobs (recommended):

1. **Color**

   ○ Use green for classic "heal", gold/white for holy, blue/purple for arcane, teal for sci-fi.

2. **Intensity / Brightness**

   ○ Increase for "ultimate" heal, decrease for subtle regen.

3. **Scale**

   ○ Small scale: potion tick / low heal

   ○ Medium: standard ability

   ○ Large: ultimate / checkpoint heal

4. **Duration / Timing**

   ○ Short burst: quick feedback for combat

   ○ Longer: ritual heal, shrine heal

5. **Spawn Rate / Particle Count**

   ○ Reduce for mobile/VR or heavy scenes

   ○ Increase for hero moments

Recommended presets (examples):

● **Holy Heal:** warm white + gold, high bloom, soft sparkles

● **Nature Heal:** green + leaf-like sparkle feel, slightly slower motion

● **Arcane Heal:** cyan/purple, runes emphasized, sharper burst

● **Sci-fi Med:** teal/white, cleaner glow, fewer organic sparkles

6) Performance Notes / Optimization

These effects are intended for real-time gameplay. Final performance depends on scene complexity and platform.

Best practices:

- Use **burst variants** for frequent combat heals (short-lived, cheap).

- Use **loop variants** sparingly (regen aura on 1–4 characters at once, not 40).

- Lower **spawn rate** and **lifetime** first when optimizing.

- Prefer attaching to a socket/bone that doesn't jitter excessively (avoid excessive motion blur look).

- If you have both CPU & GPU emitters:

    - **GPU** for dense visuals (sparkle fields)

    - **CPU** for gameplay-critical particles that need collision/events (if used)

7) Networking Notes (Multiplayer)

The pack is cosmetic only. In multiplayer:

- Replicate **the heal event** (or ability activation) in your gameplay code

- Spawn/activate the VFX on each client (or multicast spawn depending on your architecture)