THE VISUAL EFFECTS OF INFAMOUS: SECOND SON

MATT VAINIO

Senior Visual Effects Artist
Sucker Punch Productions

TEEN
T
CONTENT RATED BY
ESRB

OVERVIEW

- Designing the look and feel for Smoke and Neon powers
- The creation of a new toolset and pipeline for I:SS
- Pros and cons from our new workflow and lessons learned attempting to make something new

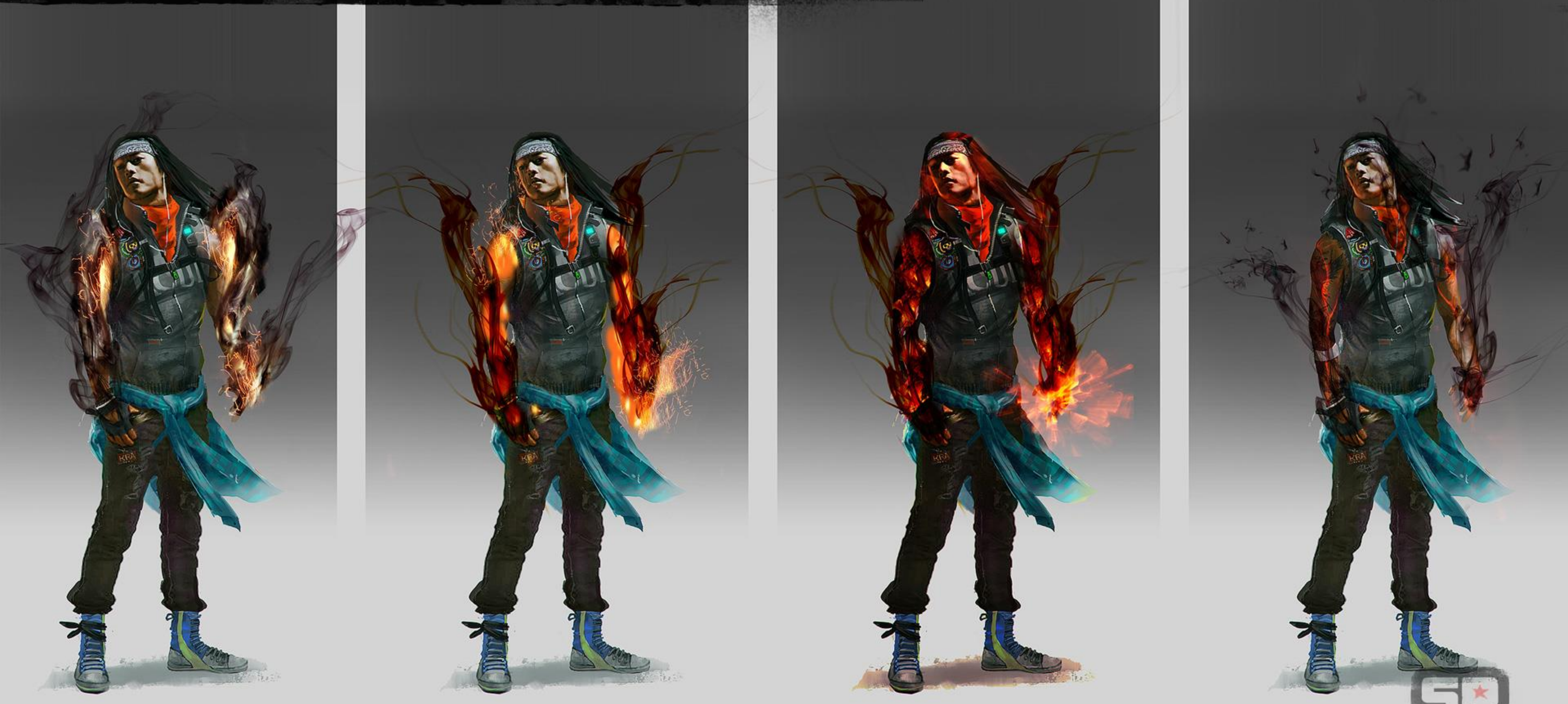
SMOKE POWERS

CASE STUDY | SMOKE POWERS

- Defining the visual language of Smoke
- The parameters of the game design
- Powers, from concept to completion
    - Smoke Dash
    - Vent Travel

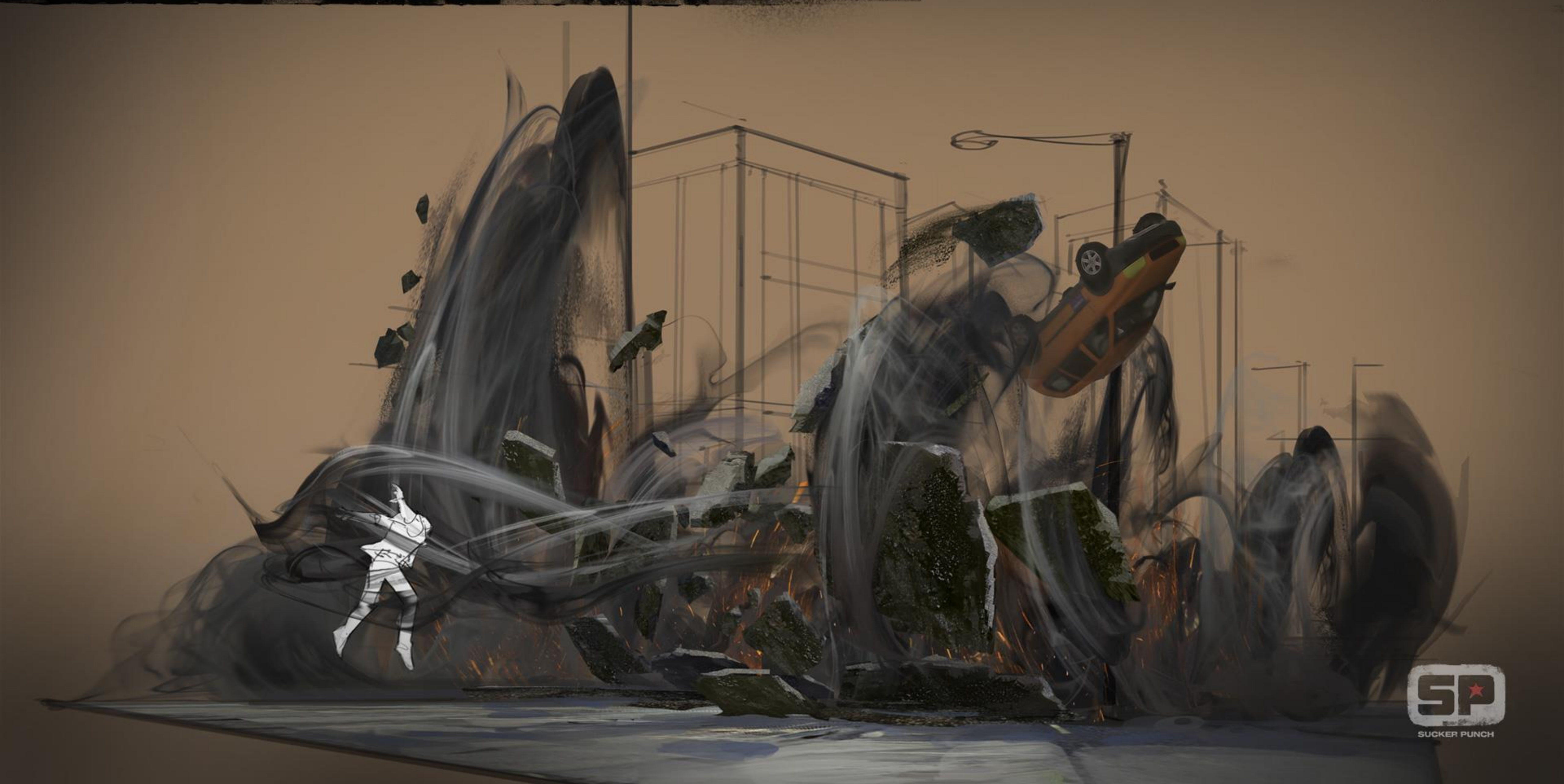
CONCEPT | SMOKE BOLT

CONCEPT | SMOKE AMBIENT



SUCKER PUNCH

CONCEPT | SMOKE ORBITAL WAVE



SMOKE | THE VISUAL LANGUAGE

- Lots of pre-production concepts
- What qualities make smoke feel real or more importantly, believable
- Reference pointed us towards these pillars
  - Wind and turbulence
  - Lighting and compositing in the scene
- Experimentation with third party pre-rendered visuals led to our creation of a real-time curl noise implementation

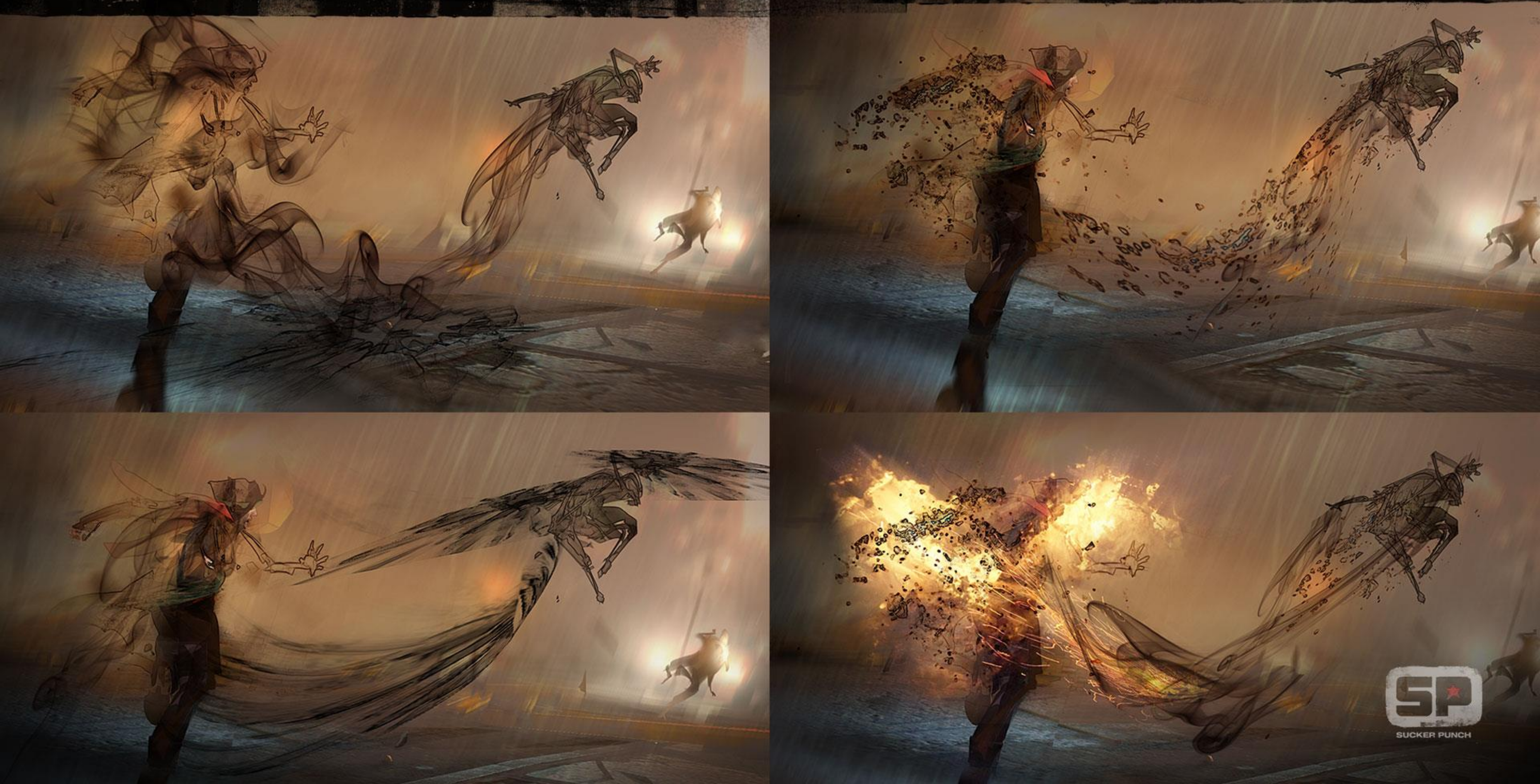
HOUDINI CURL NOISE / WIND TEST

SMOKE | THE DESIGN

- Problems:
  - Early tests had difficulty in feeling powerful
  - It was problematic to track the powers as they moved through space or see them during dark times of day and shadowed areas

- Result:
  - Smoke needed to be the contrail but not the impacting force
  - Add ash and lights to aid in visual tracking

CONCEPT | SMOKE DASH



SMOKE | DASH

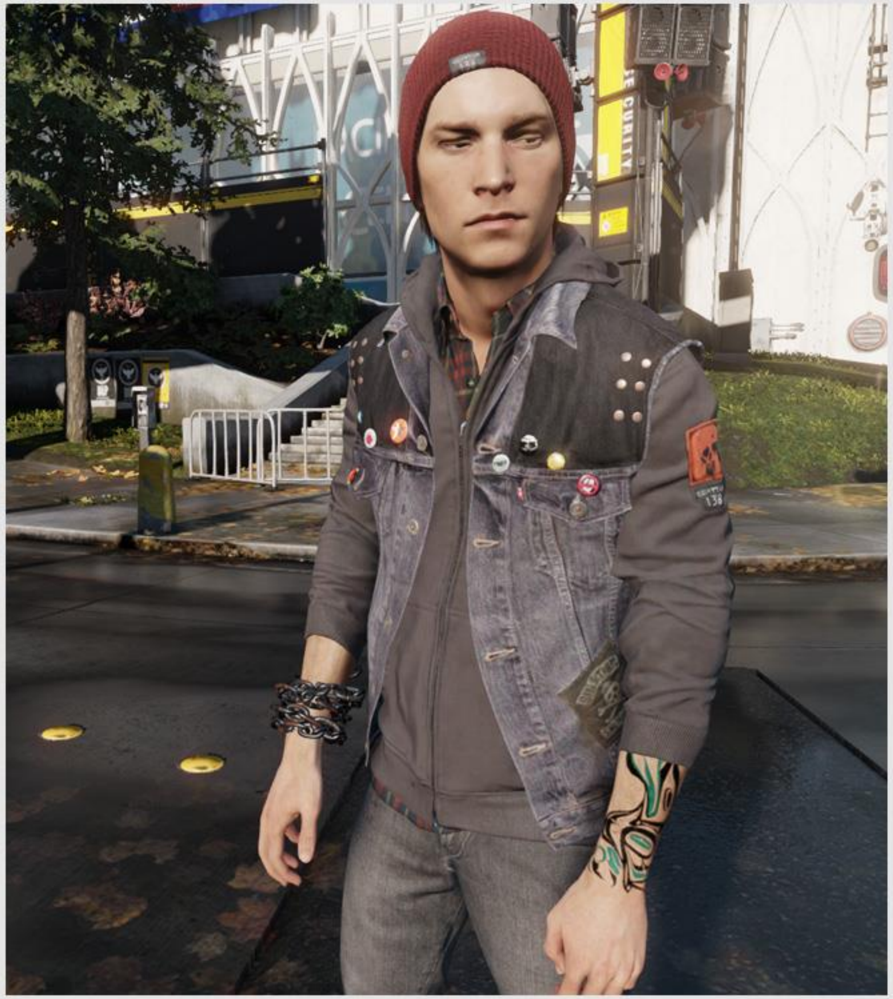
- Dematerialize hero into smoke and ash using the hero’s mesh
- The hero’s particle mesh has positional, UV, normal and color data – essentially a low-res version of the hero
- The same particles that leave from where the hero dematerialized reform into the hero on the dash exit
- Smoke ribbons spawned off the surface of the hero mesh help carry directionality of motion

SMOKE DASH

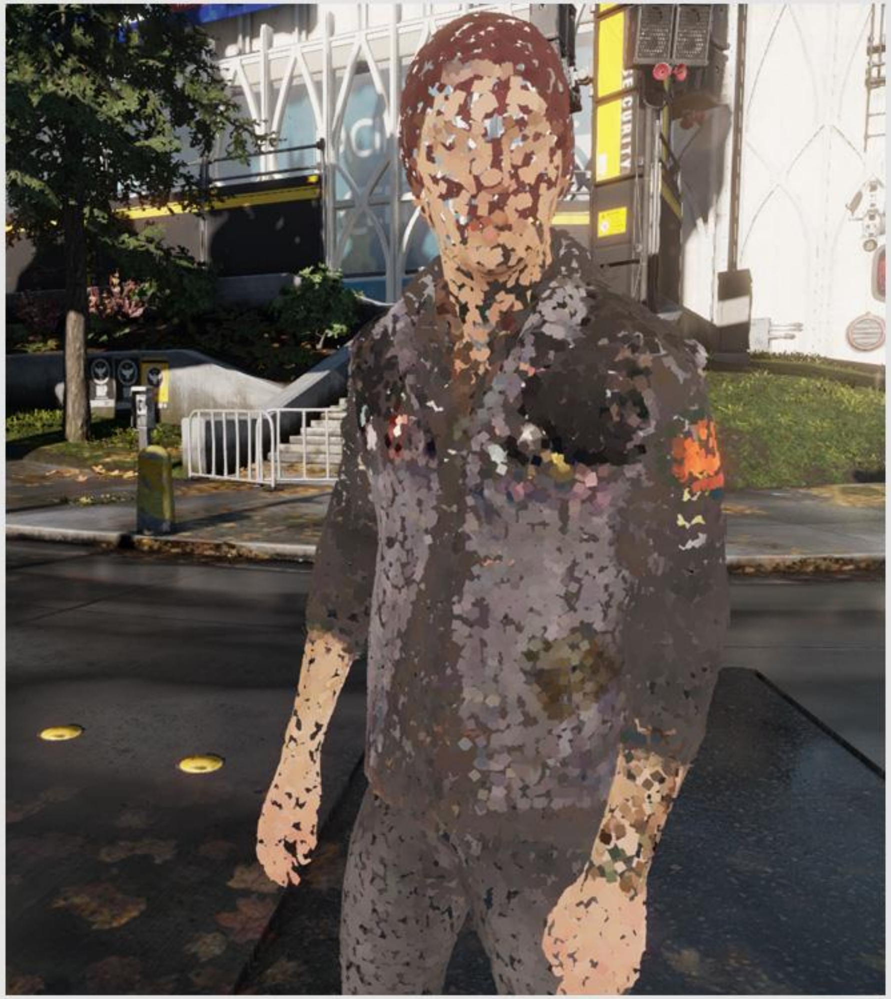
SP
SUCKER PUNCH

CENTER
SUCKER PUNCH

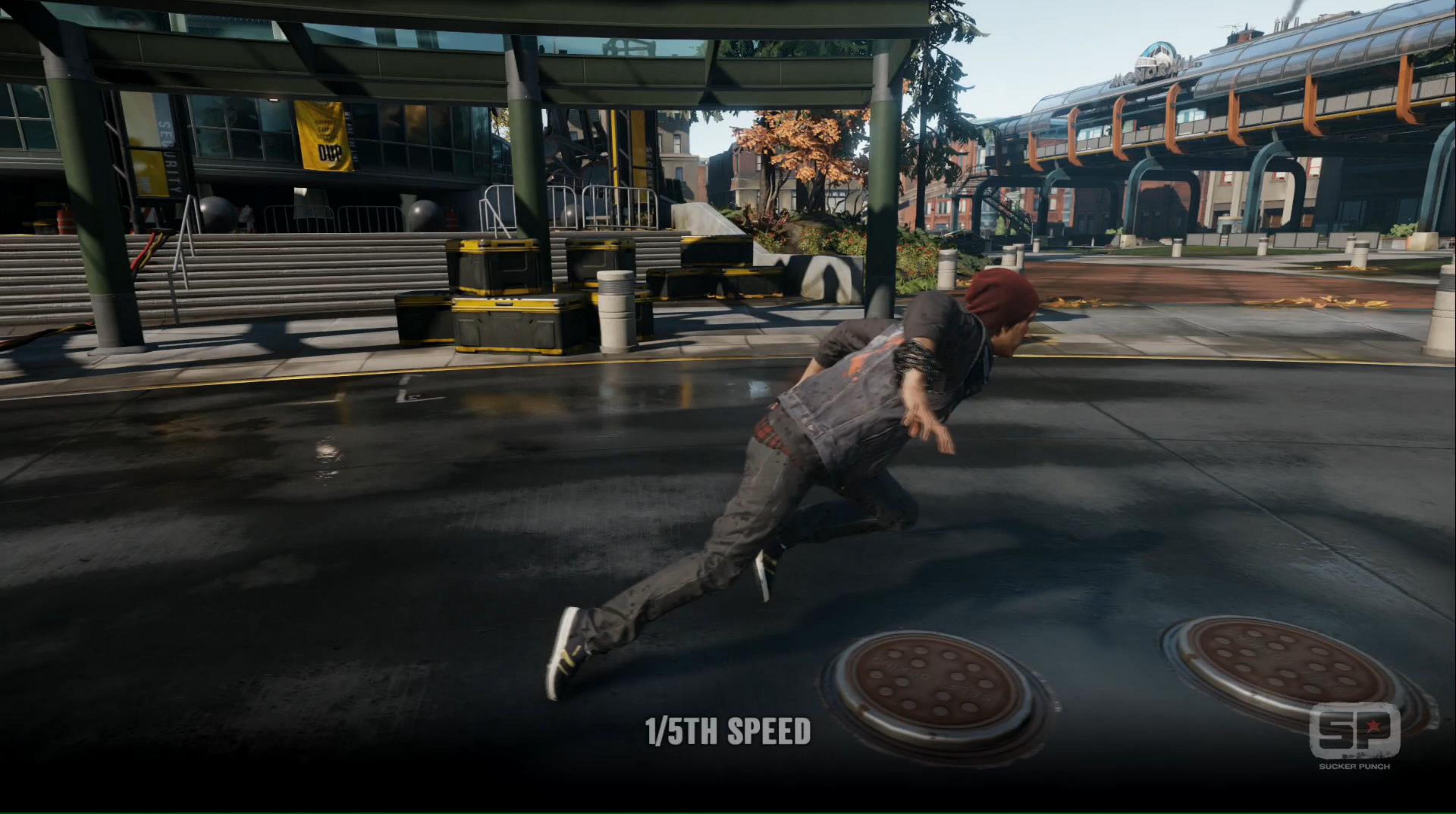
TECH | PARTICLE MESHES



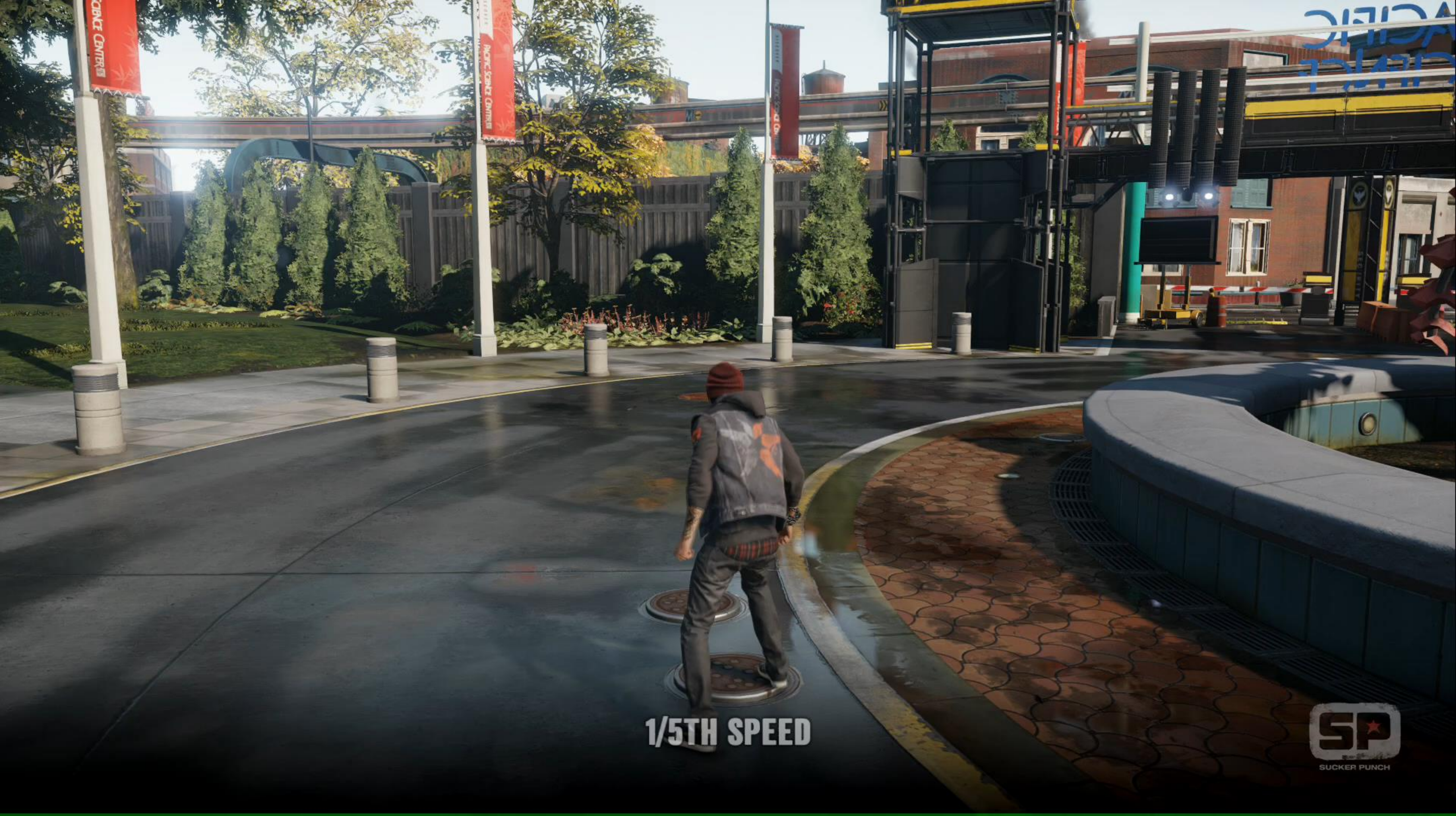
Mesh Emitter OFF



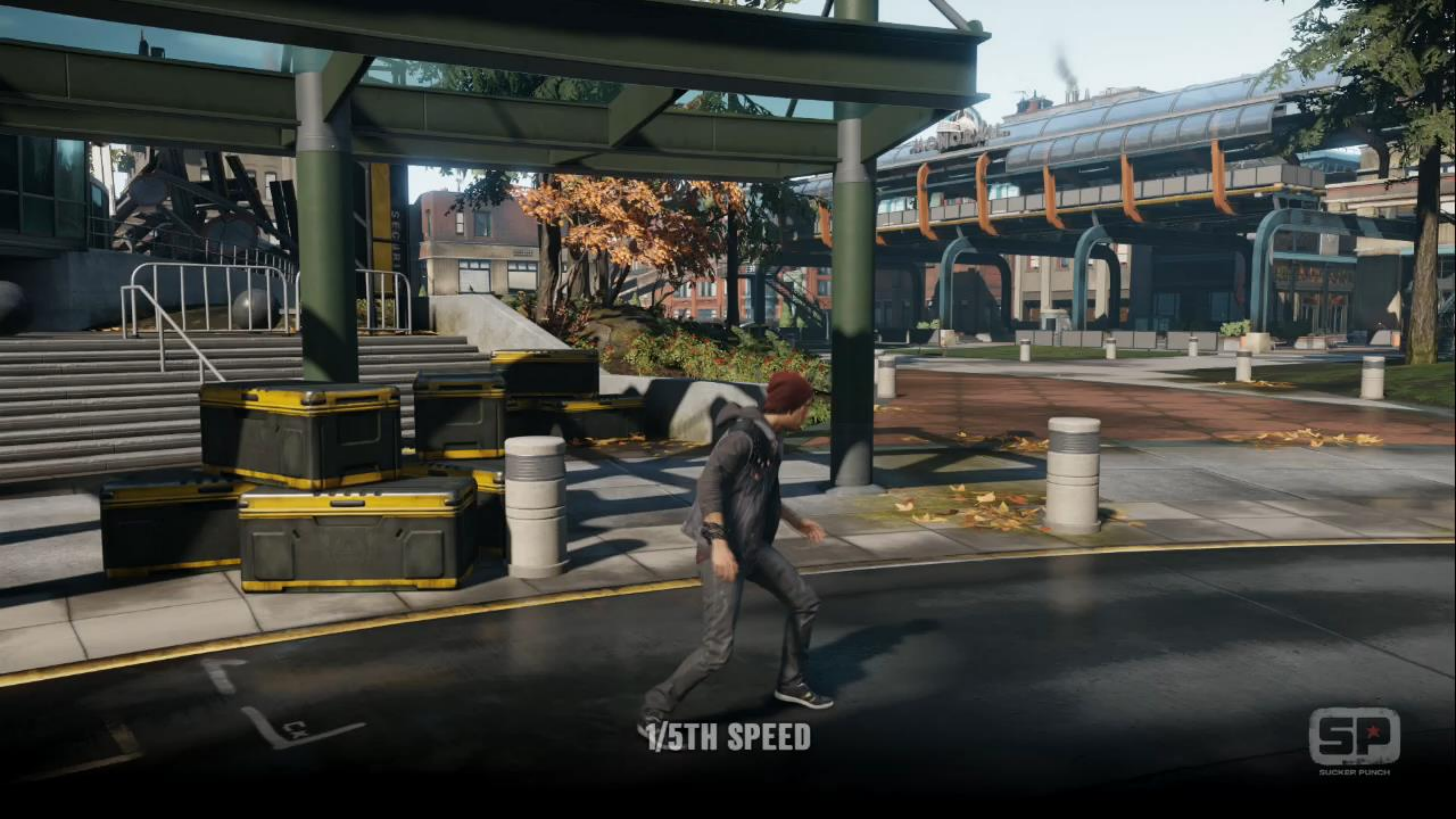
Mesh Emitter ON



1/5TH SPEED



1/5TH SPEED

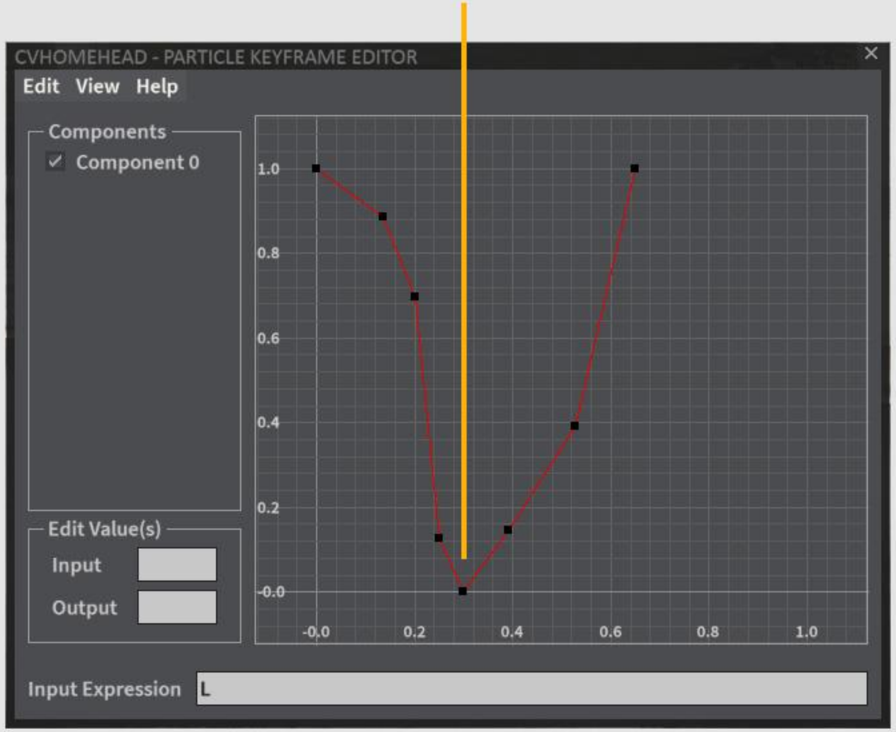


1/5TH SPEED

SMOKE | DASH

World Update

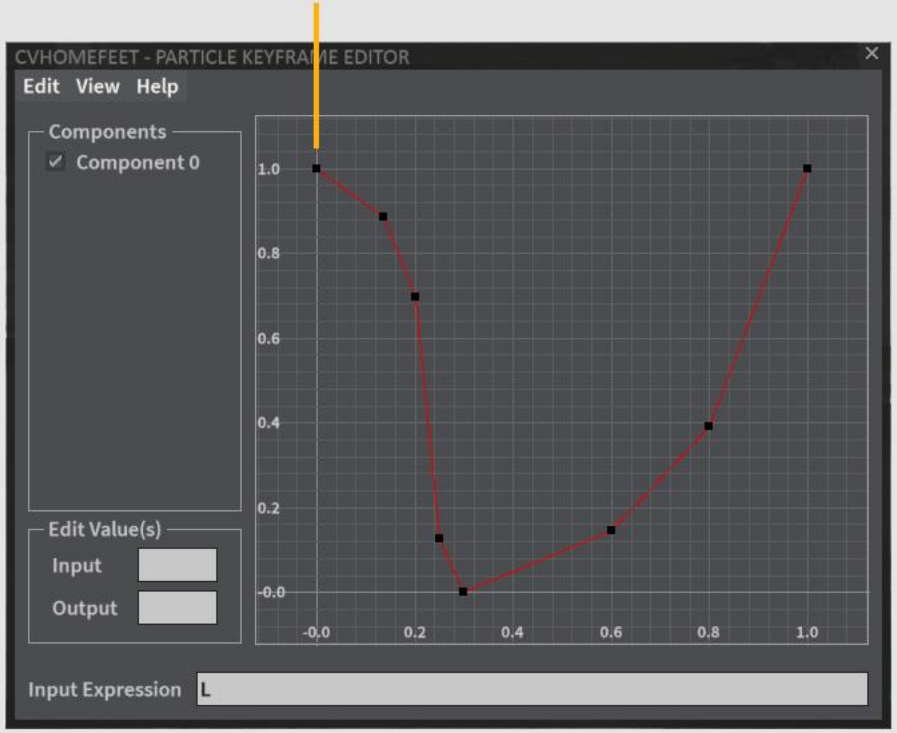
Hero Mesh Pos



CVHOMEHEAD - PARTICLE KEYFRAME EDITOR

Input Expression L

A) Particles from Head

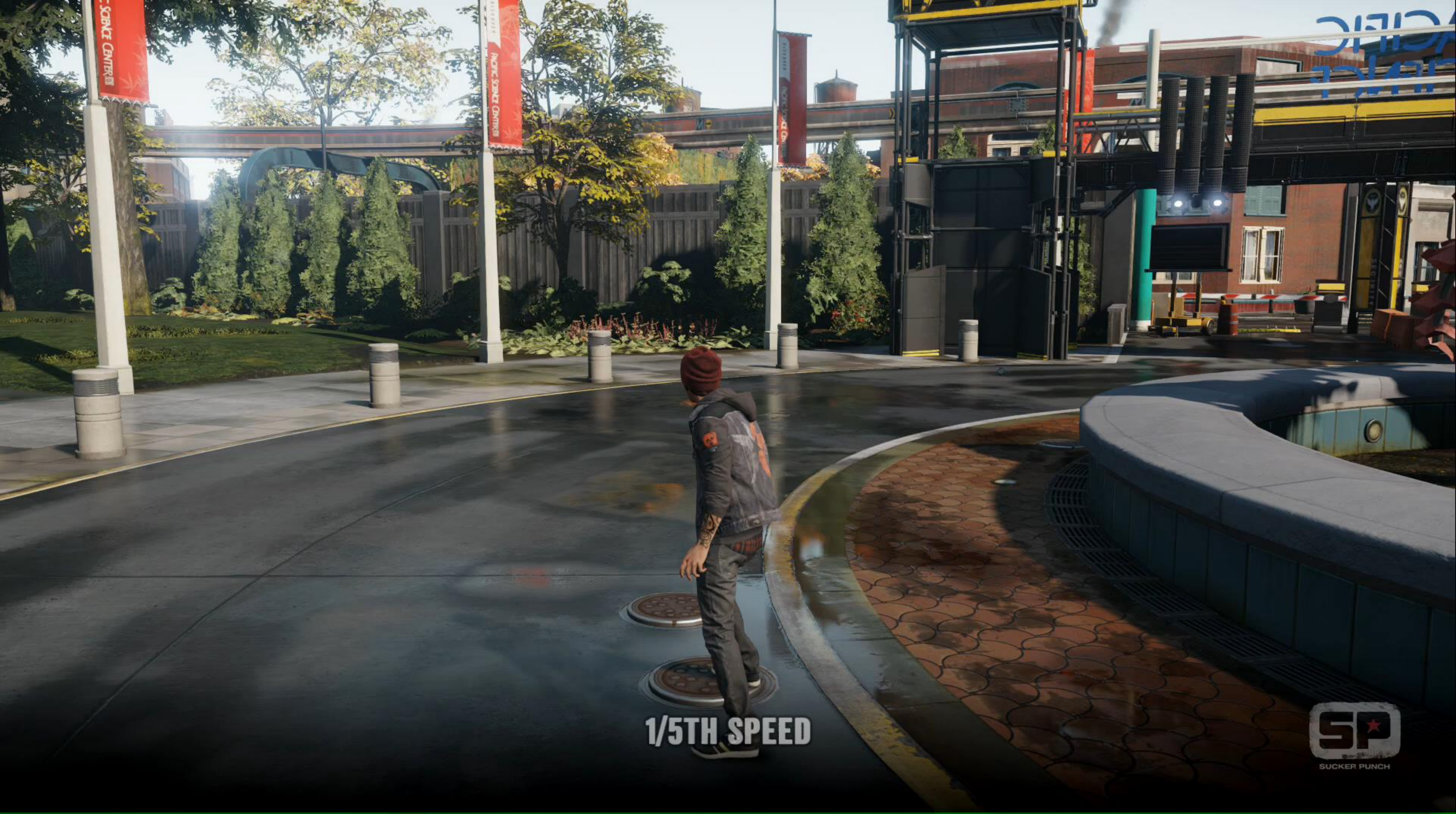


CVHOMEFEET - PARTICLE KEYFRAME EDITOR

Input Expression L

B) Particles from Feet

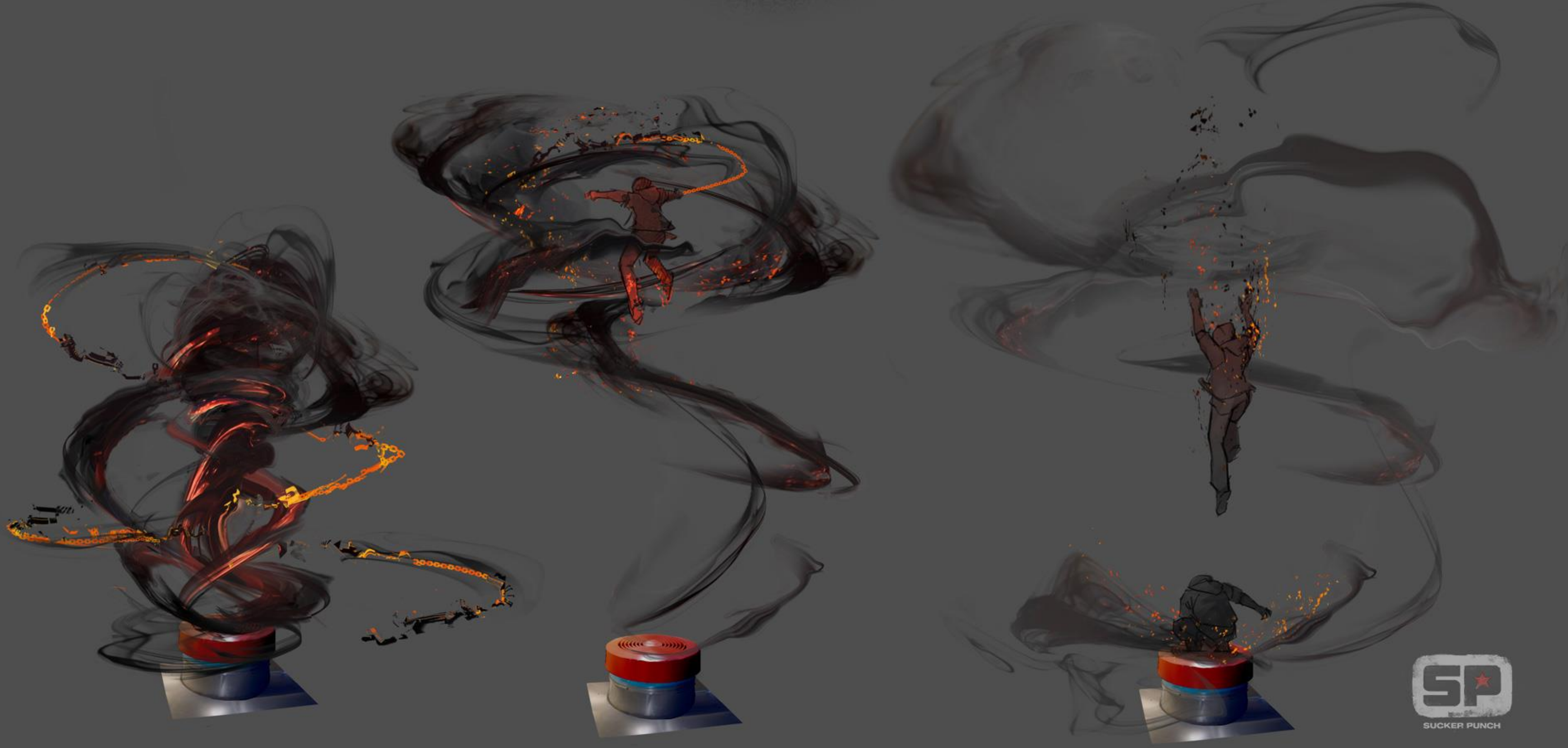
Lerp



1/5TH SPEED

LIVE AT KEXP
VOLUME NINE
PREVIEW, PURCHASE + DOWNLOAD NOW
ENTER KEXP.ORG
CITY LIMIT
Seattle
POP 621,118
Upto
EXITS 166
SUCKER PUNCH

CONCEPT | SMOKE VENT TRAVEL



SMOKE | VENT TRAVEL

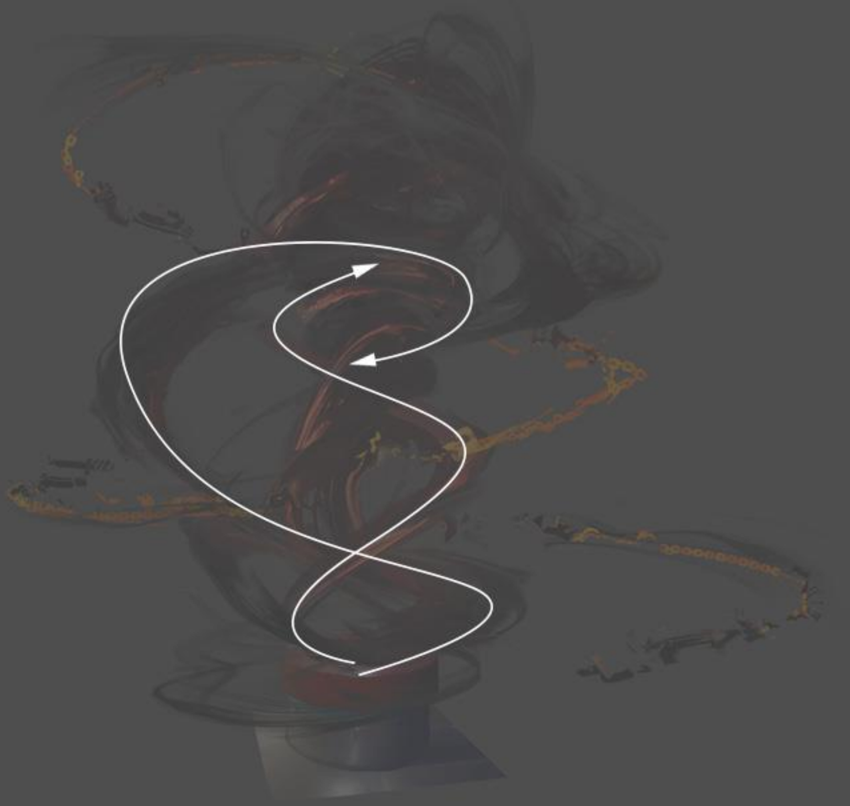
- Dematerialize hero like Smoke Dash
- Wisps created with ribbons using parent/child relationships
- Ribbon parents rotate and translate upward over their lifetime
- Ribbons utilize curl noise and a velocity vector to blow away in the wind
- Work closely with Animation team on timing

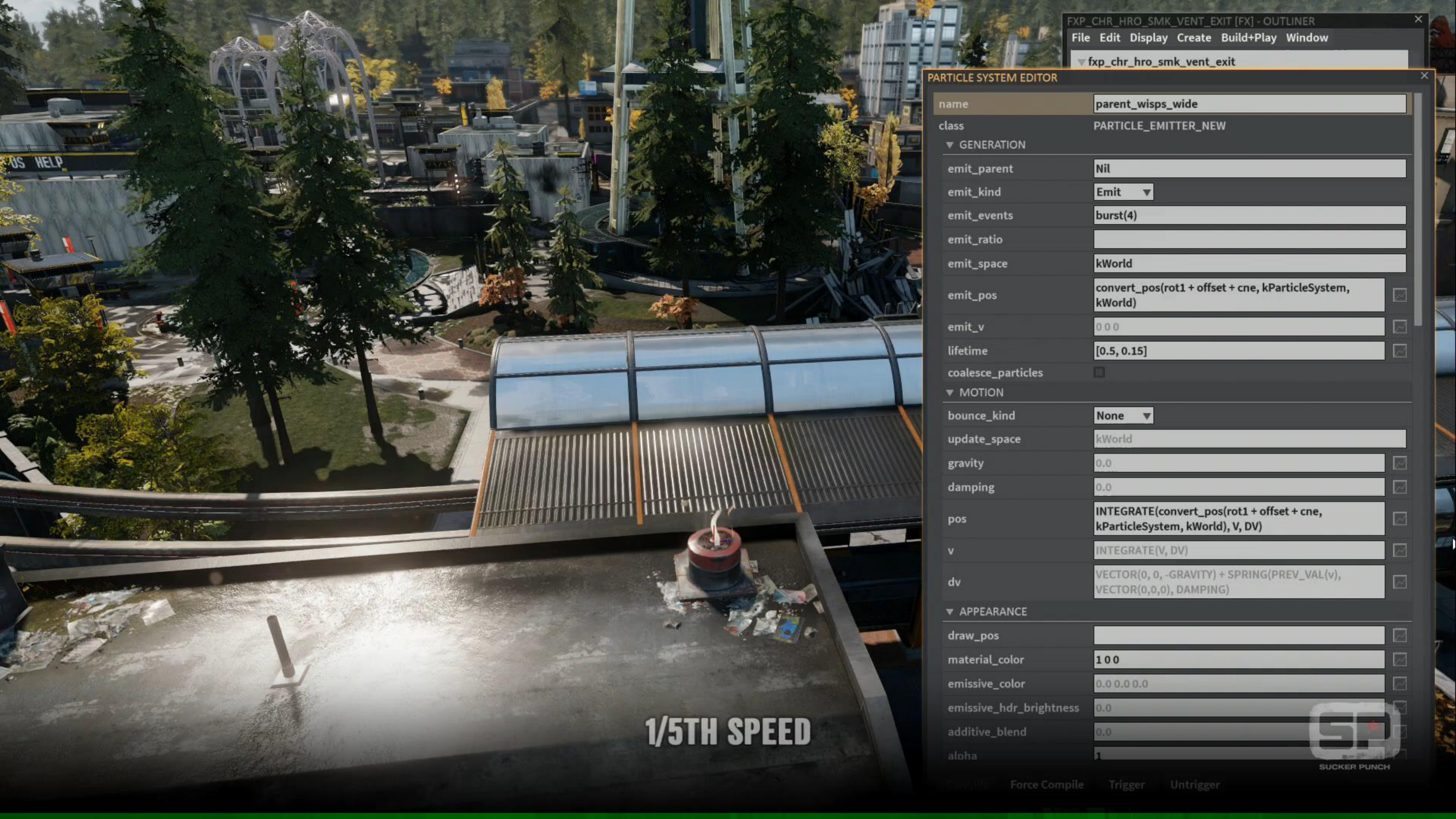
SMOKE VENT TRAVEL

SUCKER PUNCH

SP
SUCKER PUNCH

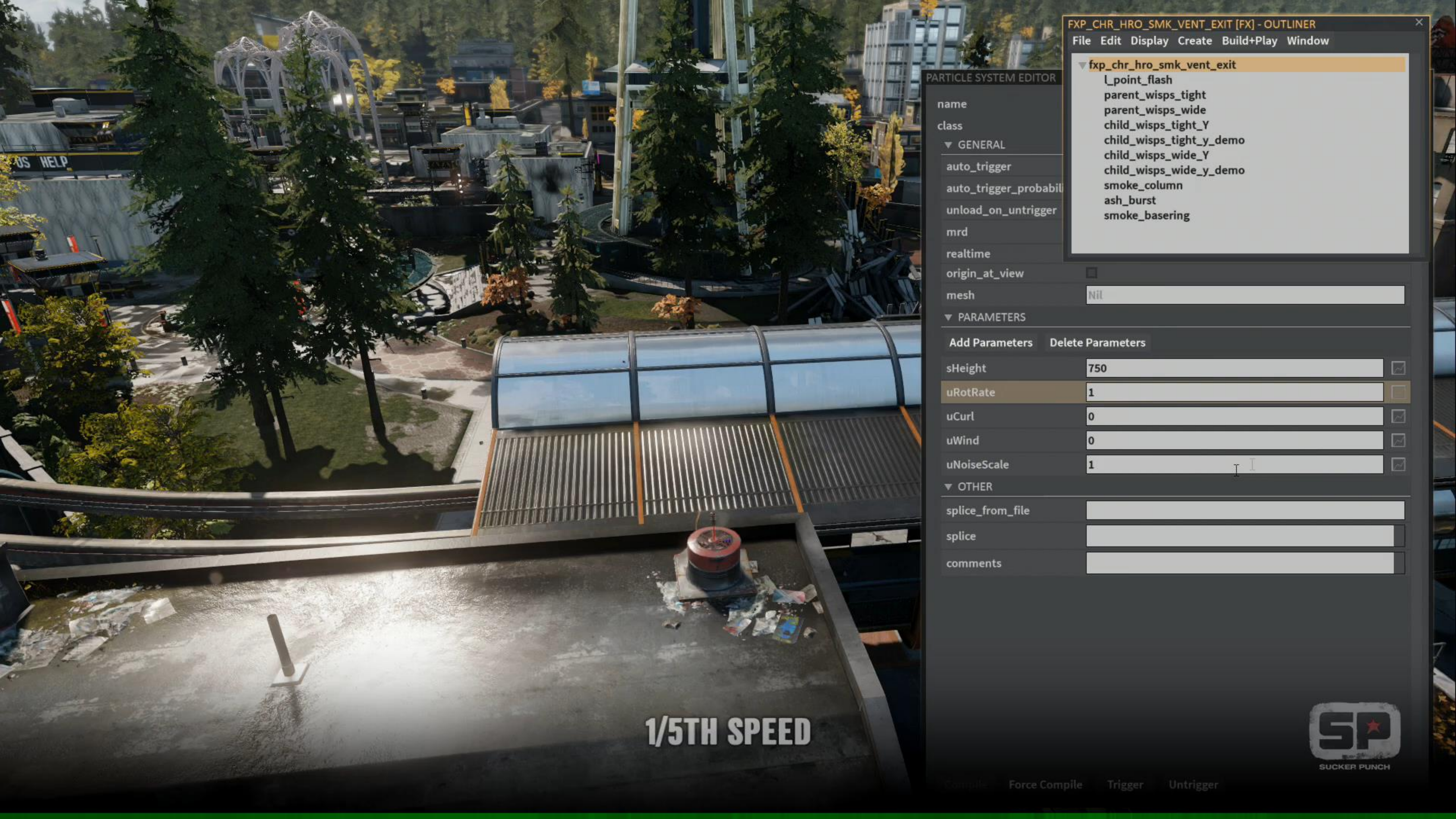
SMOKE | VENT TRAVEL





PARTICLE SYSTEM EDITOR

name	parent_wisps_wide
class	PARTICLE_EMITTER_NEW
▼ GENERATION	
emit_parent	Nil
emit_kind	Emit ▼
emit_events	burst(4)
emit_ratio	
emit_space	kWorld
emit_pos	convert_pos(rot1 + offset + cne, kParticleSystem, kWorld)
emit_v	0 0 0
lifetime	[0.5, 0.15]
coalesce_particles	□
▼ MOTION	
bounce_kind	None ▼
update_space	kWorld
gravity	0.0
damping	0.0
pos	INTEGRATE(convert_pos(rot1 + offset + cne, kParticleSystem, kWorld), V, DV)
v	INTEGRATE(V, DV)
dv	VECTOR(0, 0, -GRAVITY) + SPRING(PREV_VAL(v), VECTOR(0,0,0), DAMPING)
▼ APPEARANCE	
draw_pos	
material_color	1 0 0
emissive_color	0.0 0.0 0.0
emissive_hdr_brightness	0.0
additive_blend	0.0
alpha	1



FXP_CHR_HRO_SMK_VENT_EXIT [FX] - OUTLINER

- fxp_chr_hro_smk_vent_exit
  - l_point_flash
  - parent_wisps_tight
  - parent_wisps_wide
  - child_wisps_tight_Y
  - child_wisps_tight_y_demo
  - child_wisps_wide_Y
  - child_wisps_wide_y_demo
  - smoke_column
  - ash_burst
  - smoke_basering

PARTICLE SYSTEM EDITOR

name

class

▼ GENERAL

auto_trigger

auto_trigger_probabil

unload_on_untrigger

mrd

realtime

origin_at_view □

mesh: Nil

▼ PARAMETERS

sHeight	750
uRotRate	1
uCurl	0
uWind	0
uNoiseScale	1

▼ OTHER

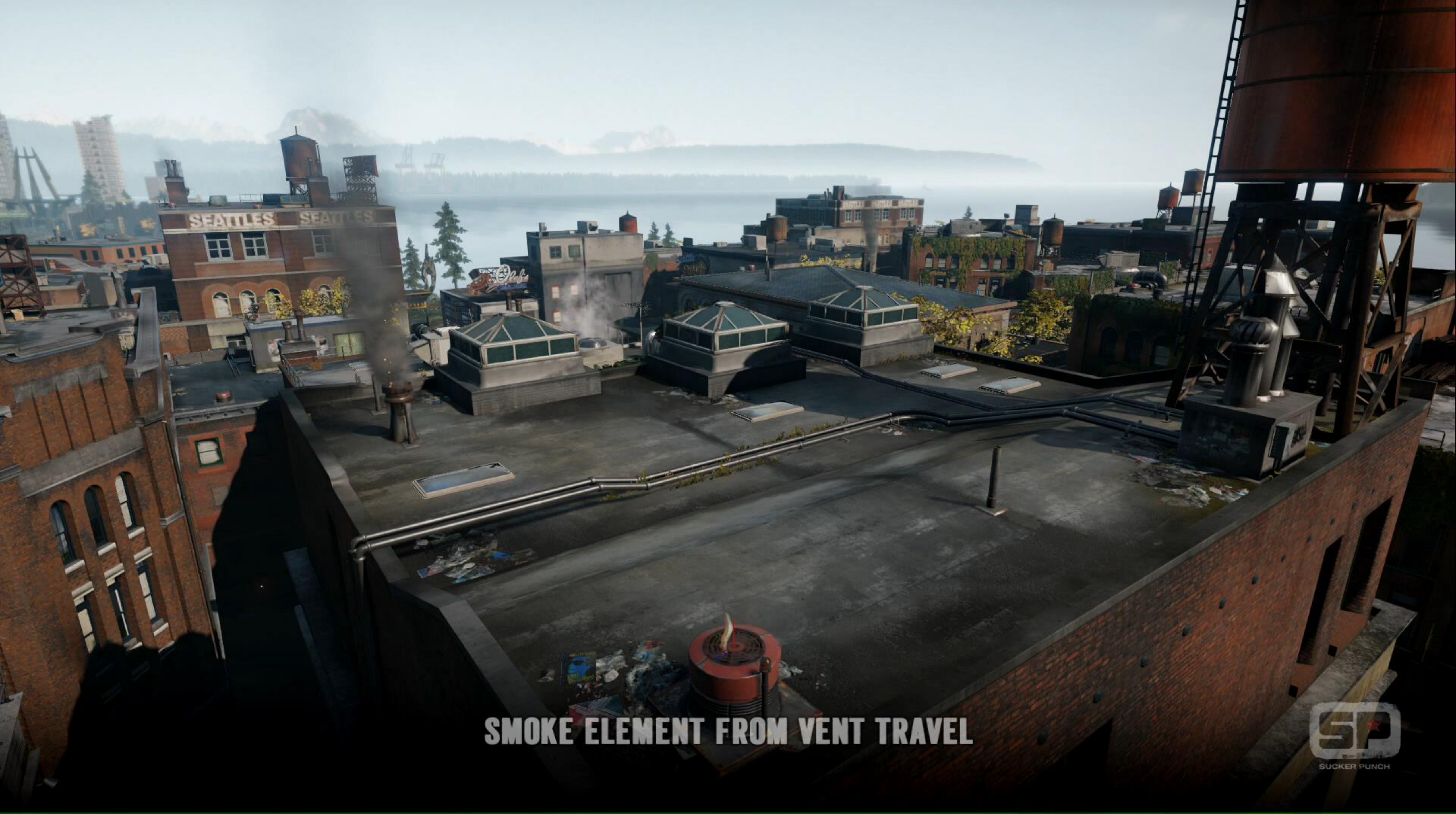
splice_from_file

splice

comments

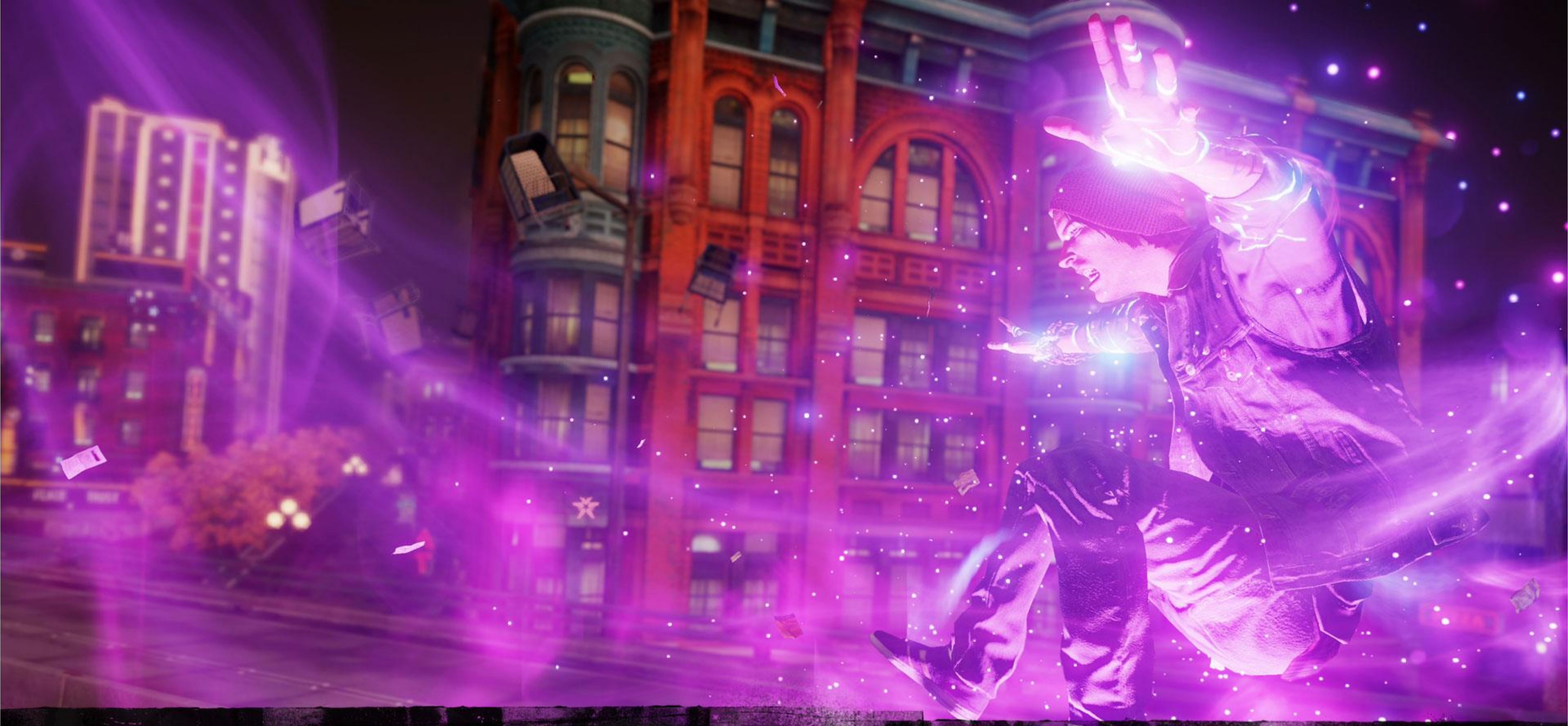
1/5TH SPEED

REAL-TIME



SMOKE ELEMENT FROM VENT TRAVEL

RMERS MA
ORGAN
SEATTLE MUSIC FESTIVAL
SUCKER PUNCH



NEON POWERS

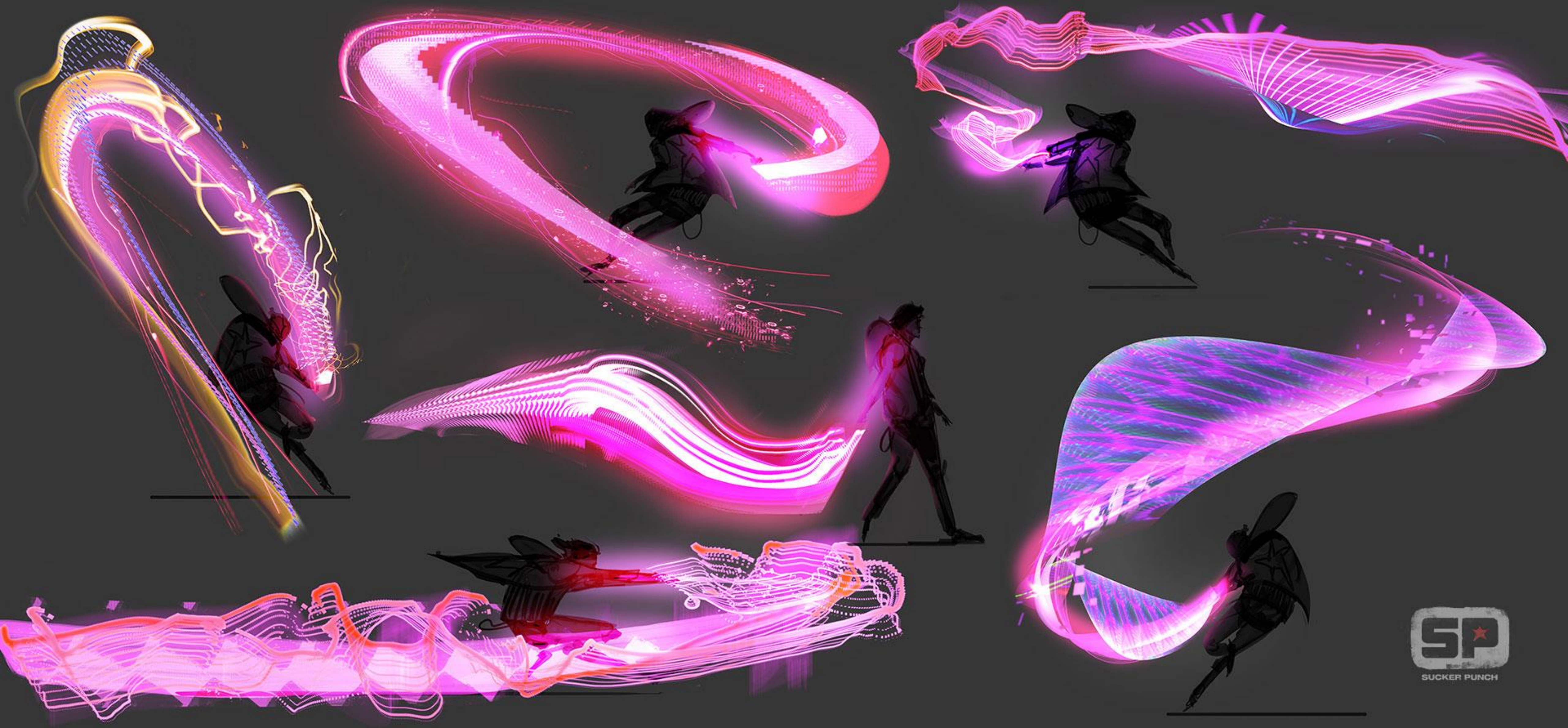
SUCKER PUNCH

CASE STUDY | NEON POWERS

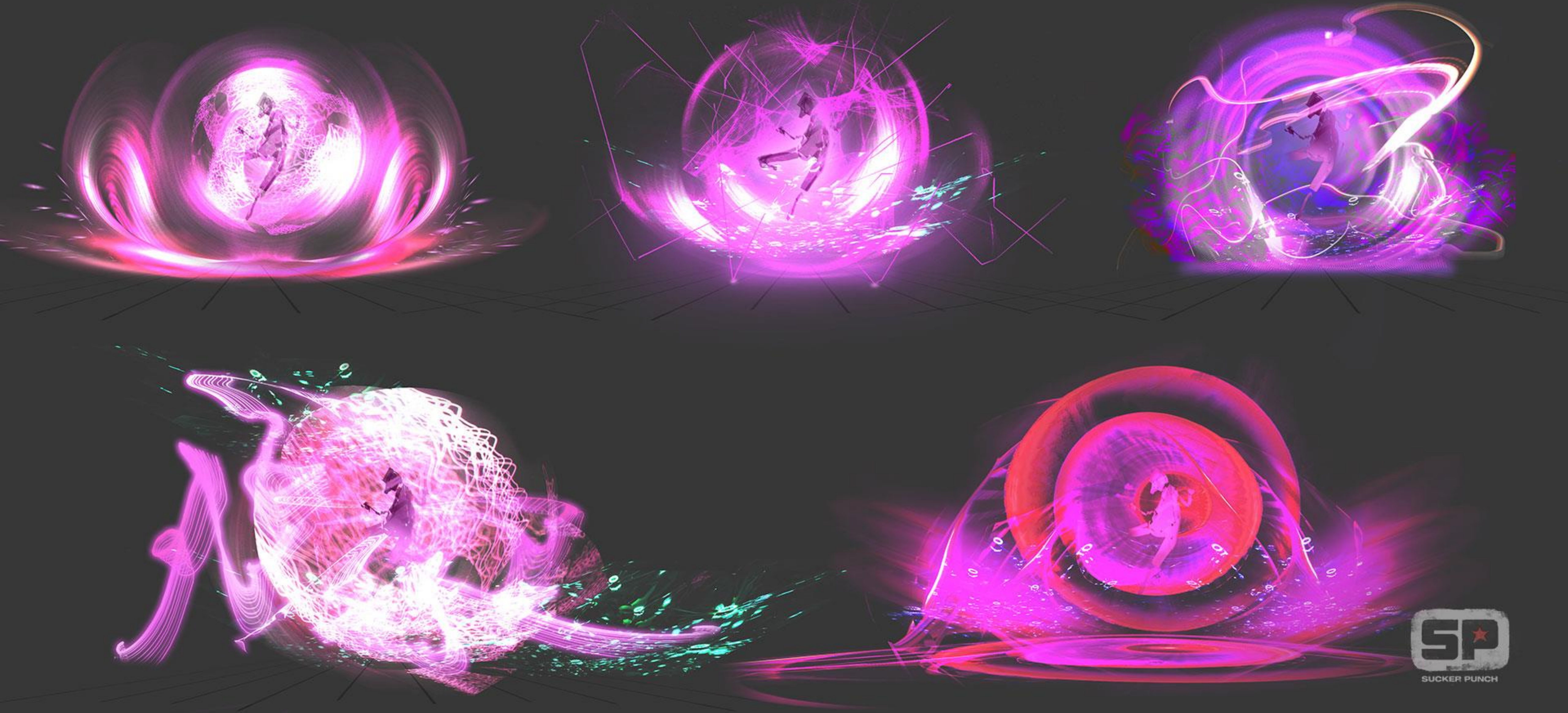
- Defining the visual language of Neon
- Powers, from concept to completion
    - Sign Drain
    - Neon Dash

SUCKER PUNCH

CONCEPT | NEON MELEE



CONCEPT | NEON SHOCKWAVE

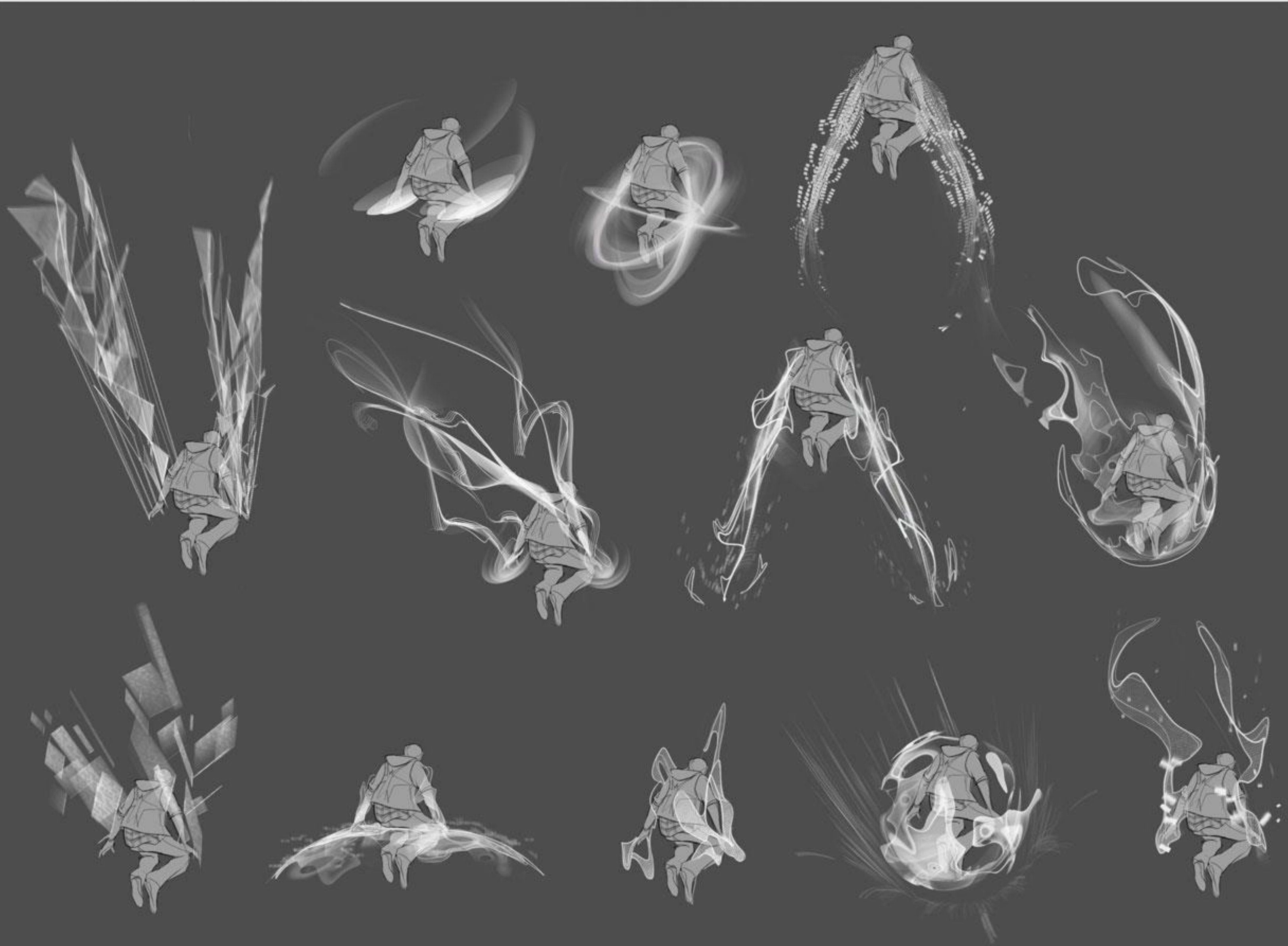


NEON | THE VISUAL LANGUAGE

- Concepts developed simultaneously with the visual effects
- Light writing complex lingering shapes
- Casting lights
- Neon as a plasma
- Utilize curl noise in different ways than smoke

CONCEPT | NEON FLOAT STUDIES

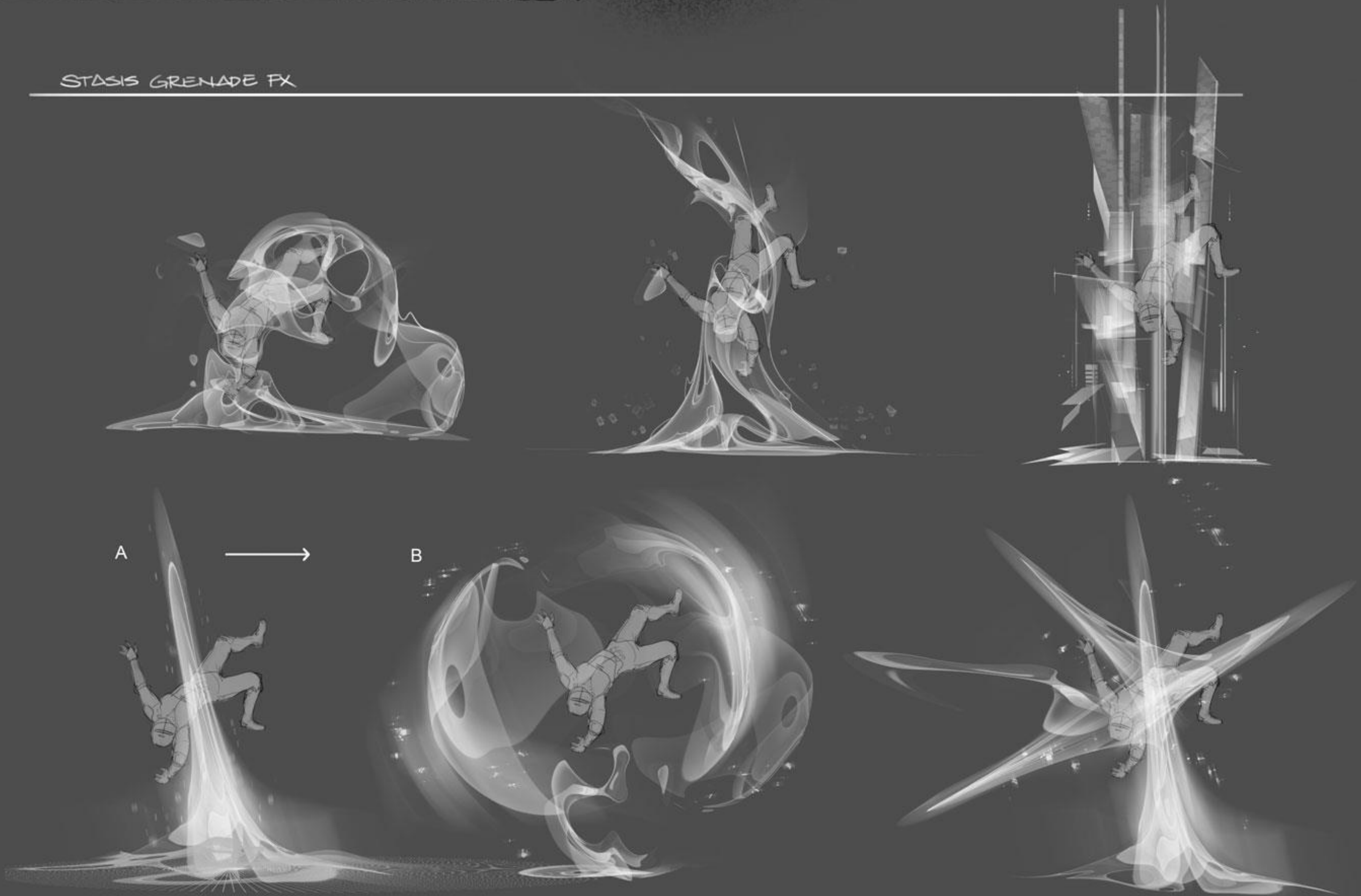
LASER FLOAT TREATMENTS



NEON | FLOAT

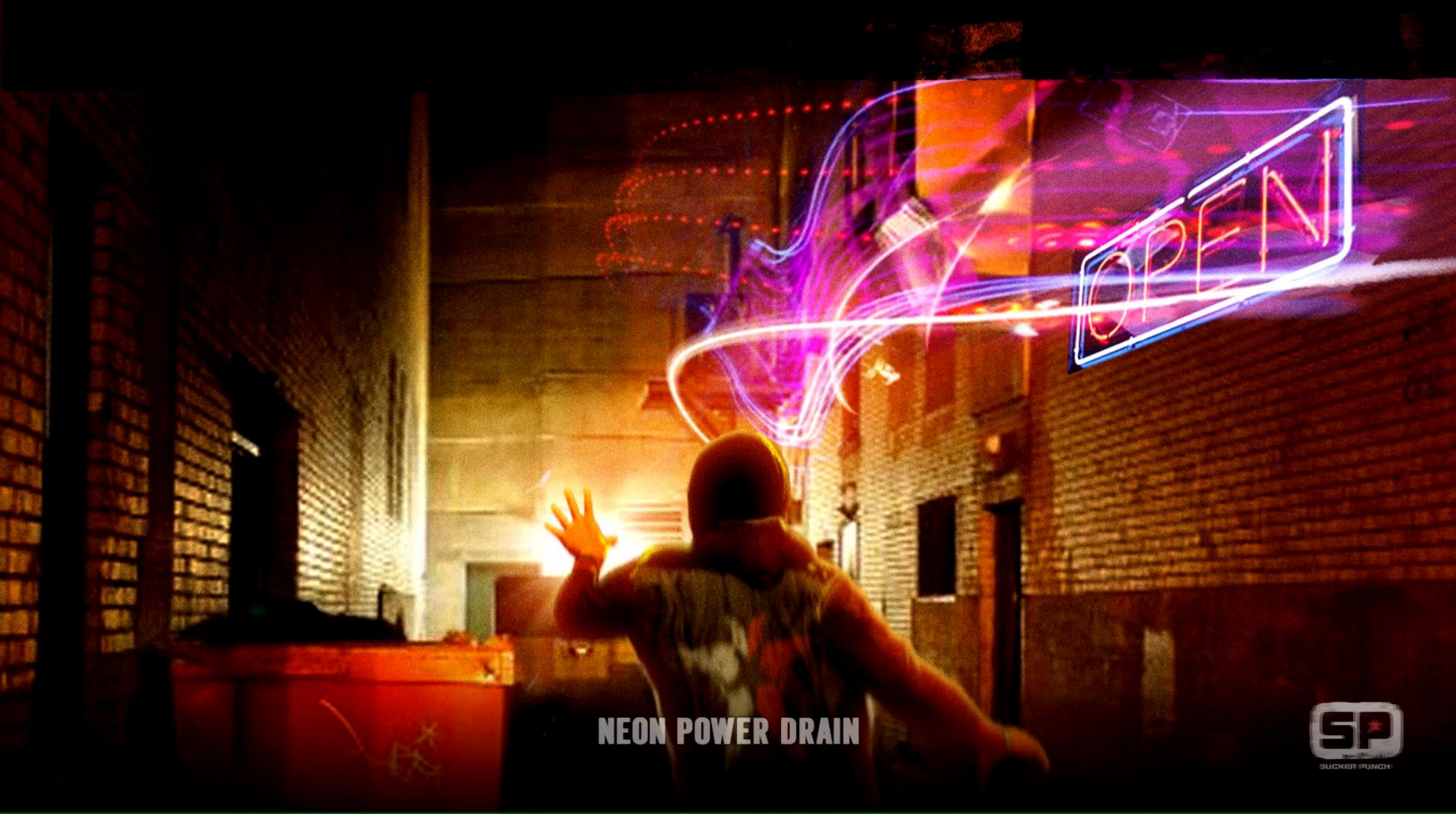
BREWERY
SEATTLES
SEATTLES
SEATTLES
SEATTLES
SP
SUCKER PUNCH

CONCEPT | STASIS GRENADE STUDIES



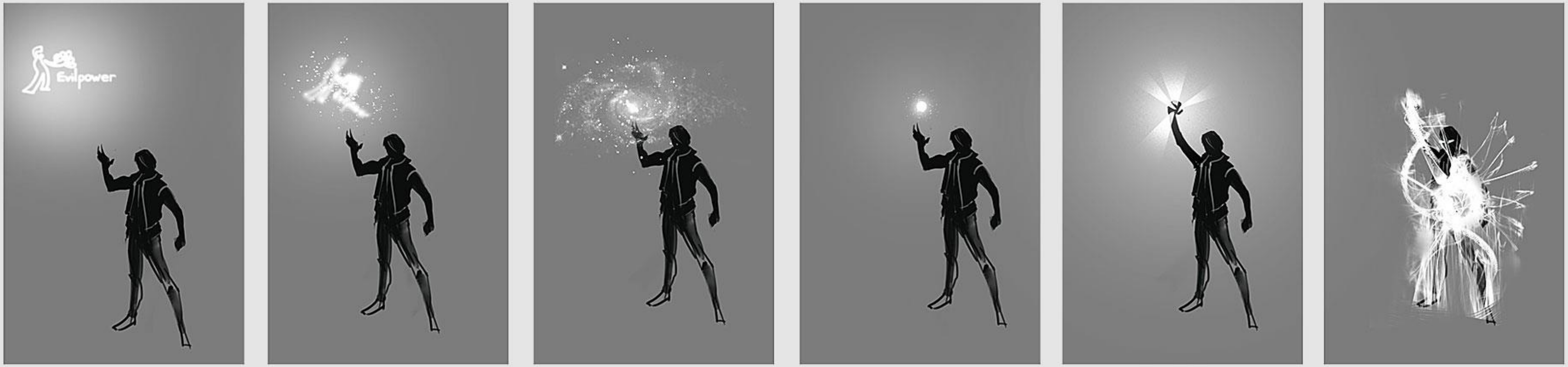
NEON | DRAIN

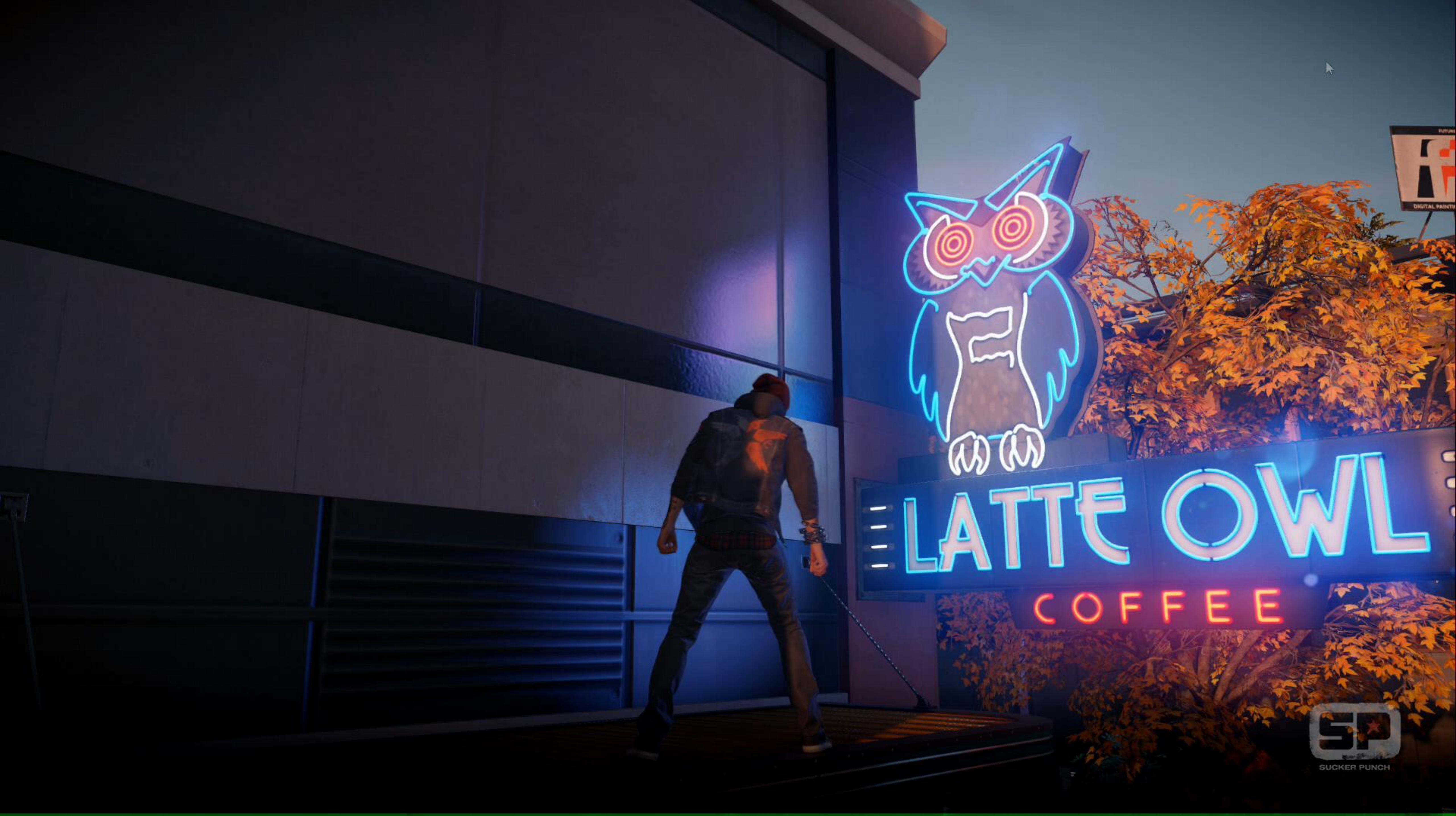
- Design goal: Drain neon particles from an in-game neon sign.
  - Uses a variant of the hero particle mesh tech
- Particles attempt to spawn only at areas above a specific brightness value
- Grabs the color from the valid position and passes to the particle system
- Accelerates towards a swirling galaxy near the hero’s hand before condensing into a single point of light



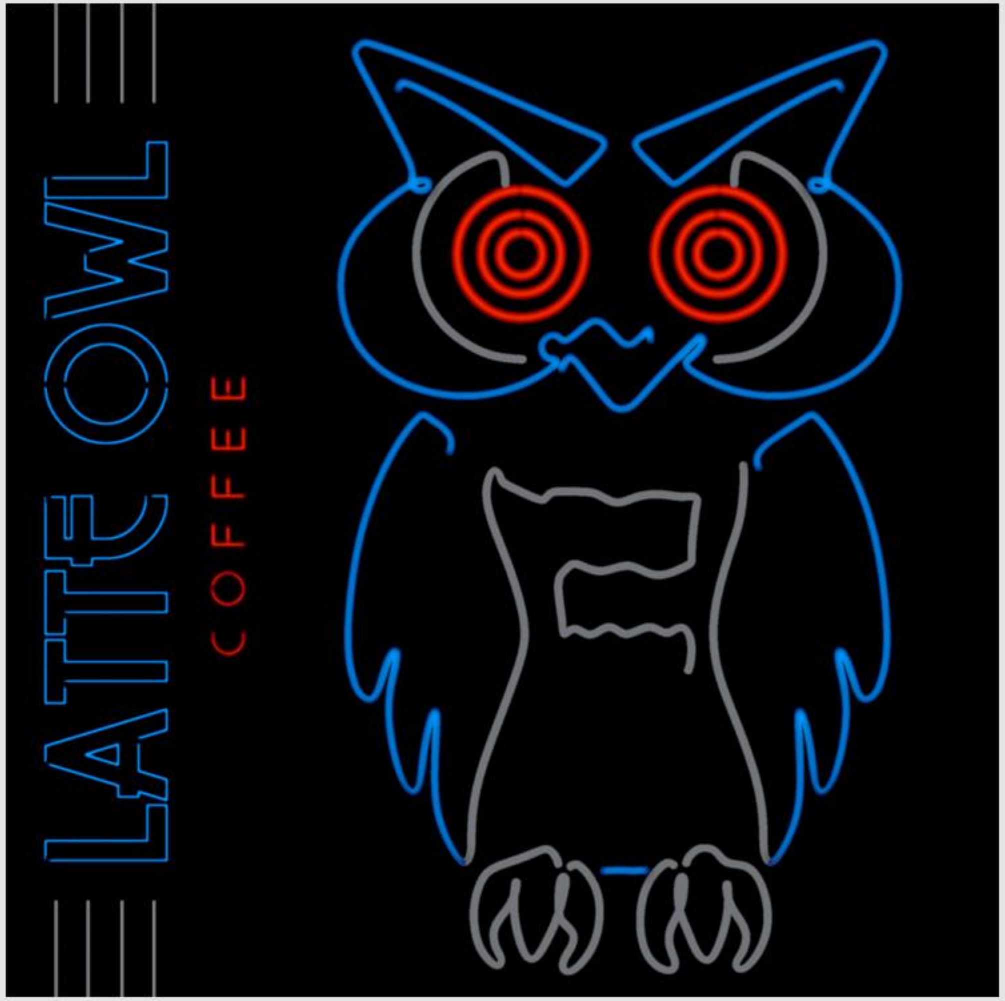
NEON POWER DRAIN

CONCEPT | NEON DRAIN

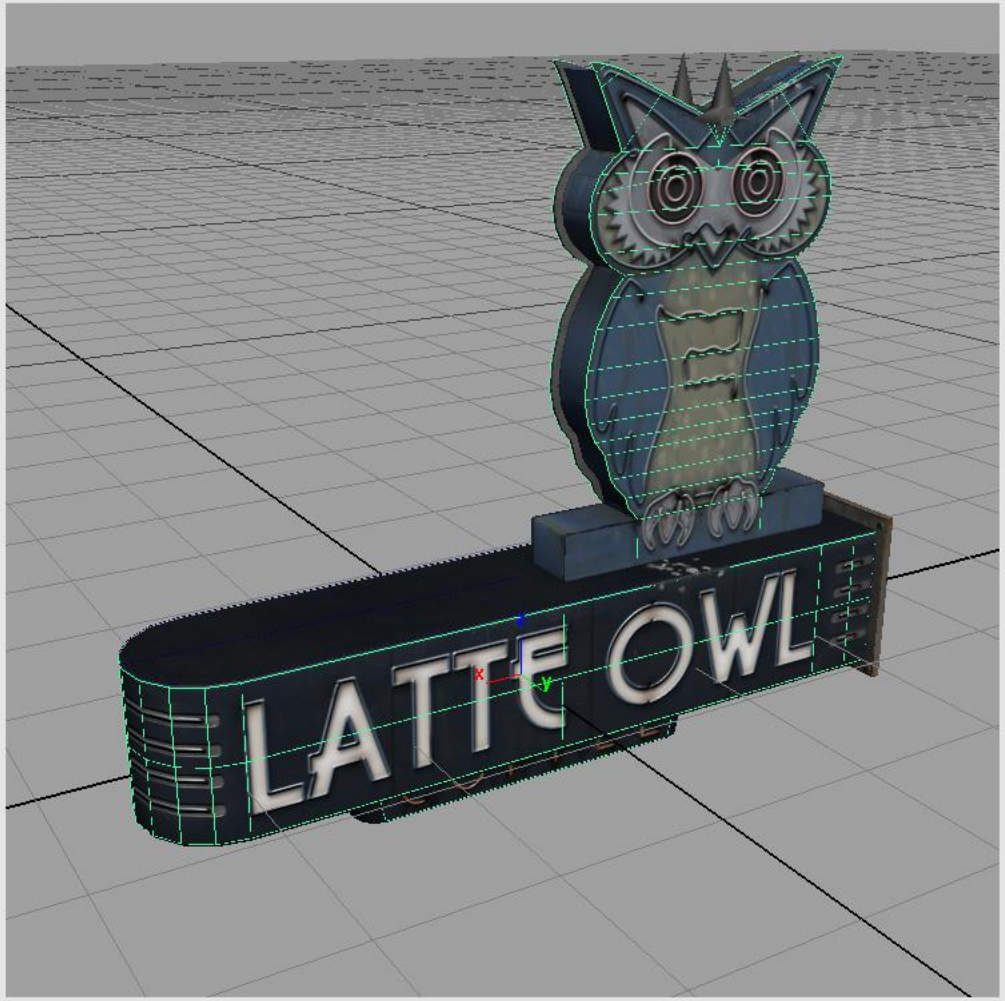



LATTE OWL
COFFEE
SUCKER PUNCH

TECH | PARTICLE MESHES



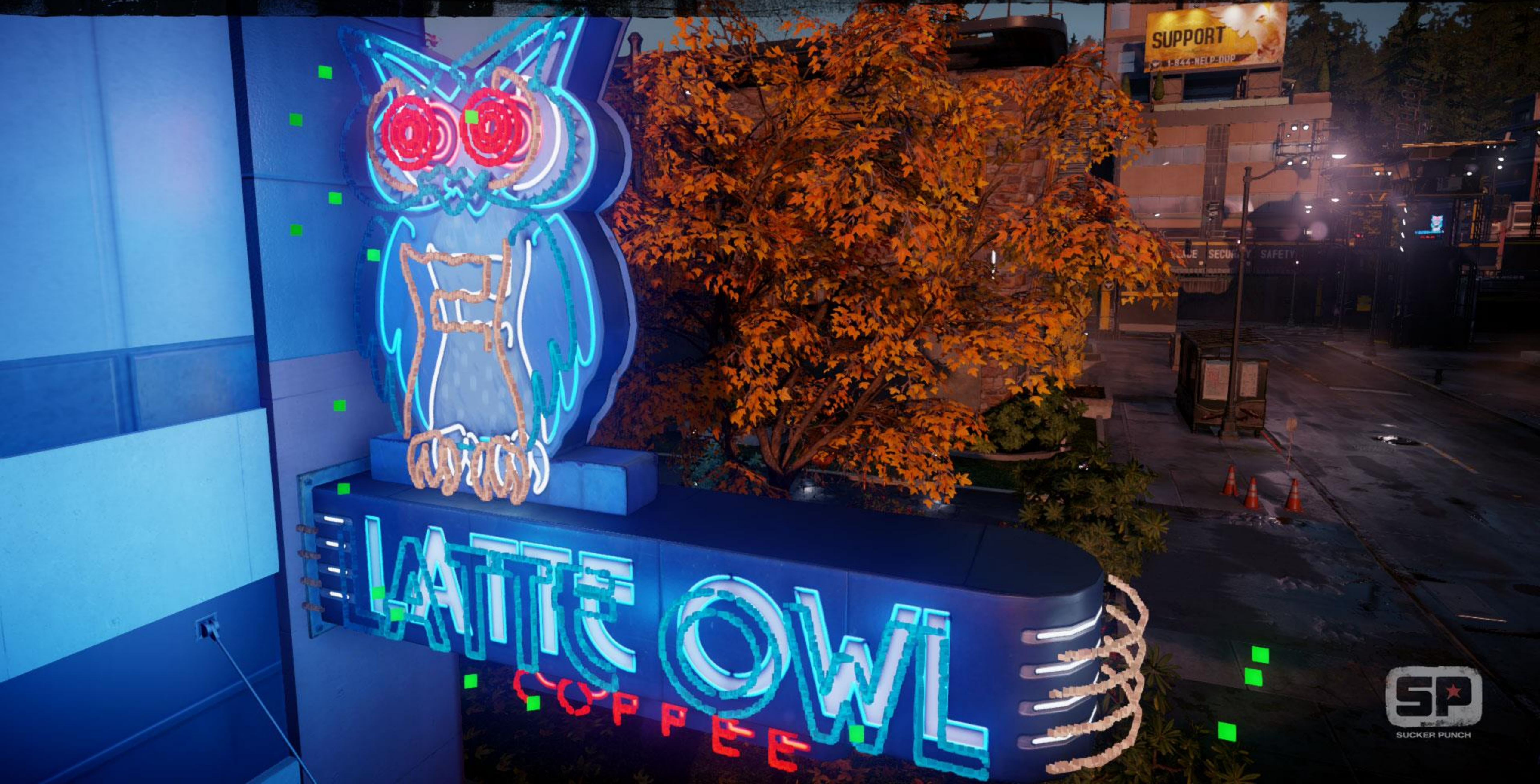
Emissive Texture



Sign Particle Mesh

FUTUREPOLY
DIGITAL PAINTING
LATTE OWL
COFFEE
1/5TH SPEED
SUCKER PUNCH

TECH | PARTICLE MESHES



LATTE OWL
COFFEE
BAR
SP
SUCKER PUNCH

NEON | DASH

- Break the hero down into a strobing silhouette
- Leave behind a lingering light-writing trail
- Body crawls with energy on exit
- Work with variable run distances
    - You can run for infinity!

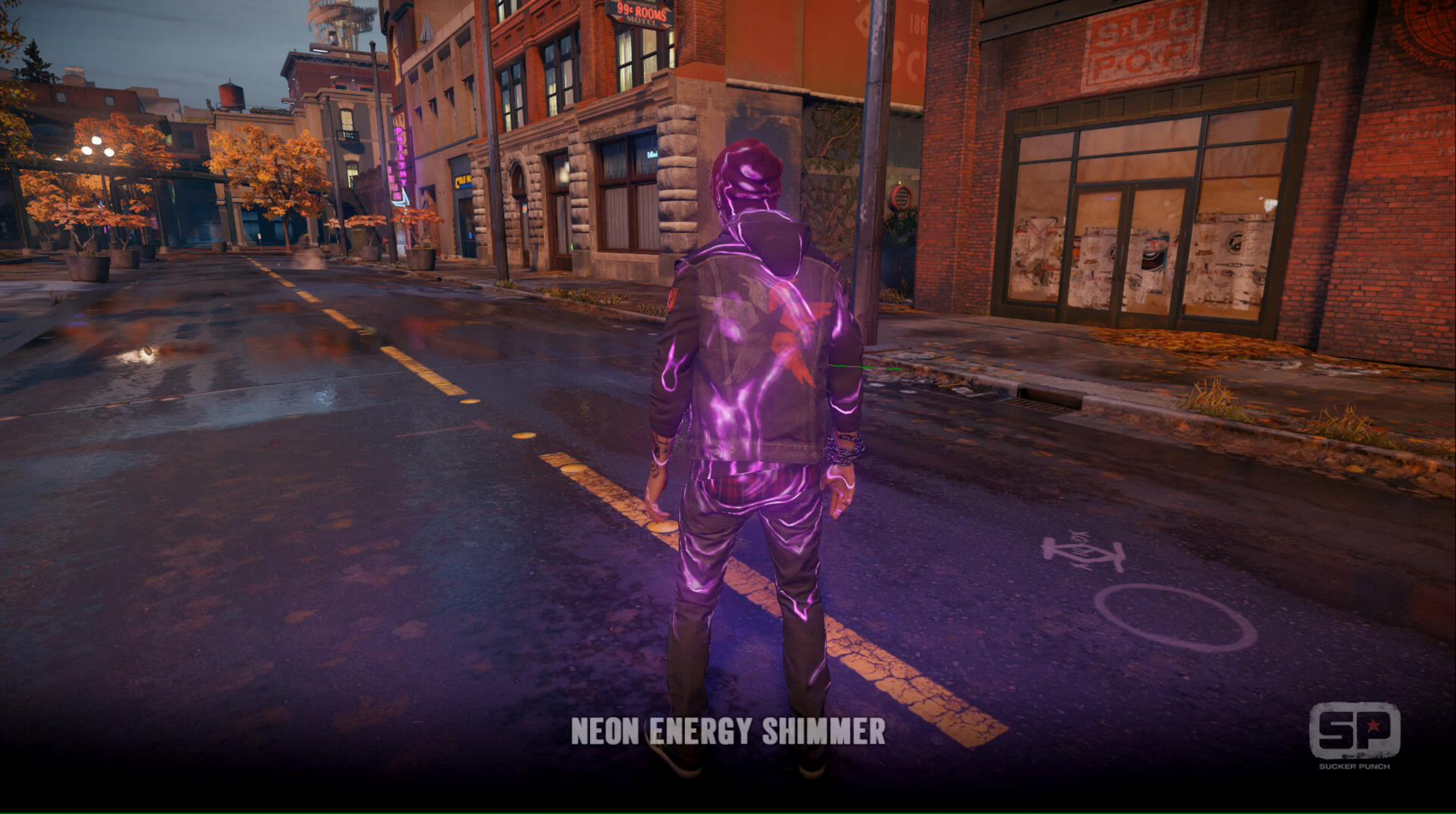
CONCEPT | NEON DASH

SUCKER PUNCH

NEON DASH

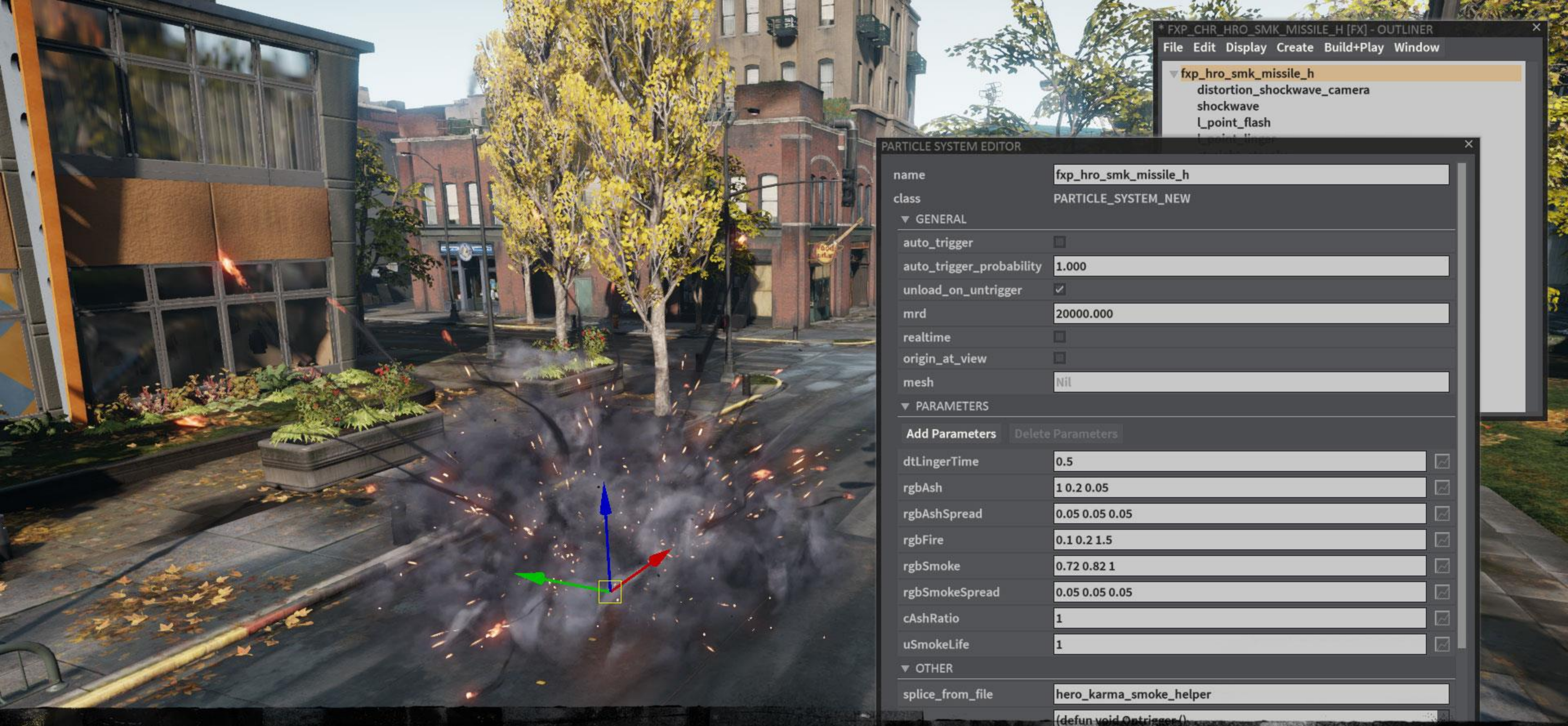
Used Again
Thrift Store
nîte
Billiards
NITE BAR
ELEPHANT
Super
CAR WA
Seattle's best car wash
SUCKER PUNCH

SUCKER PUNCH



NEON ENERGY SHIMMER

LET US HELP
SUCKER PUNCH



* FXP_CHR_HRO_SMK_MISSILE_H [FX] - OUTLINER

- fxp_hro_smk_missile_h
  - distortion_shockwave_camera
  - shockwave
  - l_point_flash
  - l_point_linger

PARTICLE SYSTEM EDITOR

name	fxp_hro_smk_missile_h
class	PARTICLE_SYSTEM_NEW
▼ GENERAL	
auto_trigger	□
auto_trigger_probability	1.000
unload_on_untrigger	☑
mrd	20000.000
realtime	□
origin_at_view	□
mesh	Nil
▼ PARAMETERS	
dtLingerTime	0.5
rgbAsh	1 0.2 0.05
rgbAshSpread	0.05 0.05 0.05
rgbFire	0.1 0.2 1.5
rgbSmoke	0.72 0.82 1
rgbSmokeSpread	0.05 0.05 0.05
cAshRatio	1
uSmokeLife	1
▼ OTHER	
splice_from_file	hero_karma_smoke_helper

```
(defun void Ontrigger()
  (Set[illegible]Smoke)
```

NEW TOOLSET

TOOLSET | CHALLENGES

- What we knew the game design required
- What we wanted to achieve artistically
- What production changes we wanted
- What does it mean to be ‘next-gen’ on new hardware?

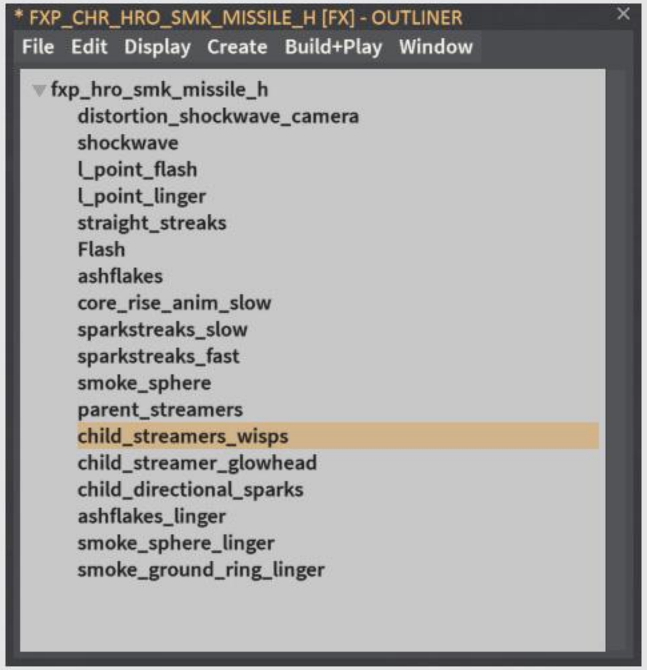
SUCKER PUNCH

TOOLSET | DECISIONS

- Create an expression based system with user parameters
- Accurate lighting/sorting/shading, integrated well into the game world
- Real-time* editing
  - Constant value adjustments
  - Keyframe editor
- Run on gpu allowing more complex expressions/simulations

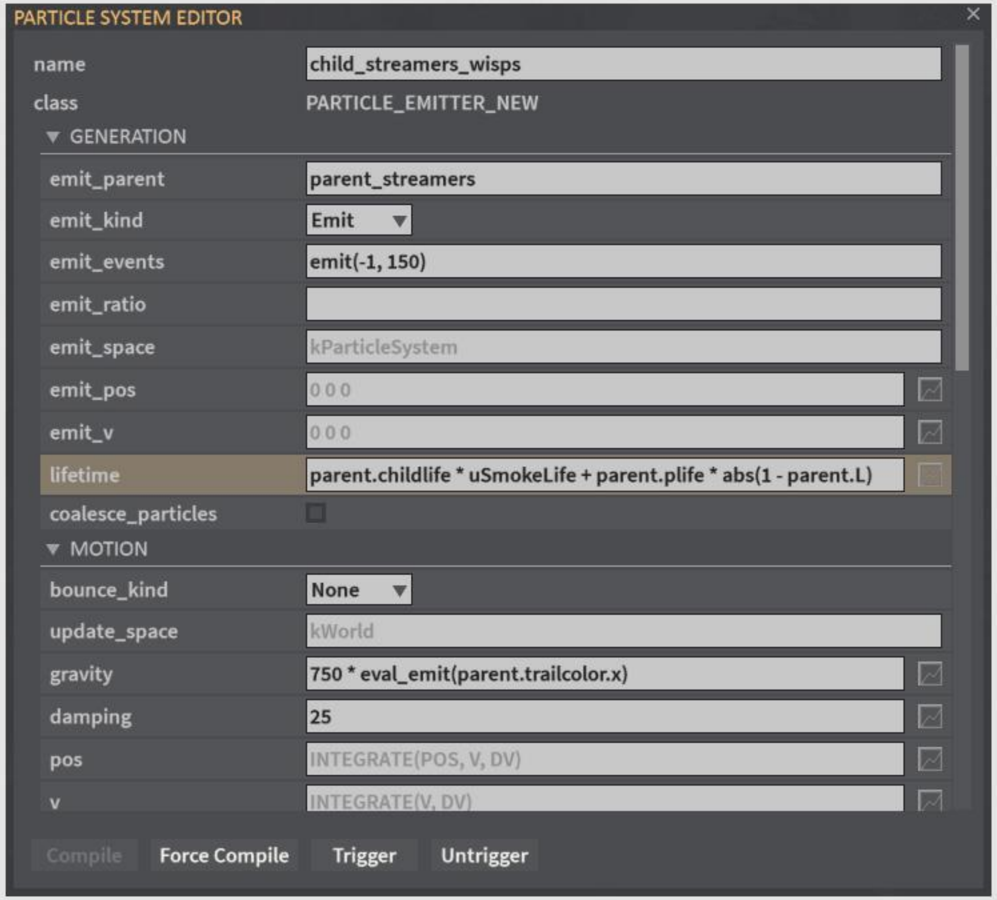
* Minor compiles necessary when changing math functions

TOOLSET | PARTICLE EDITOR



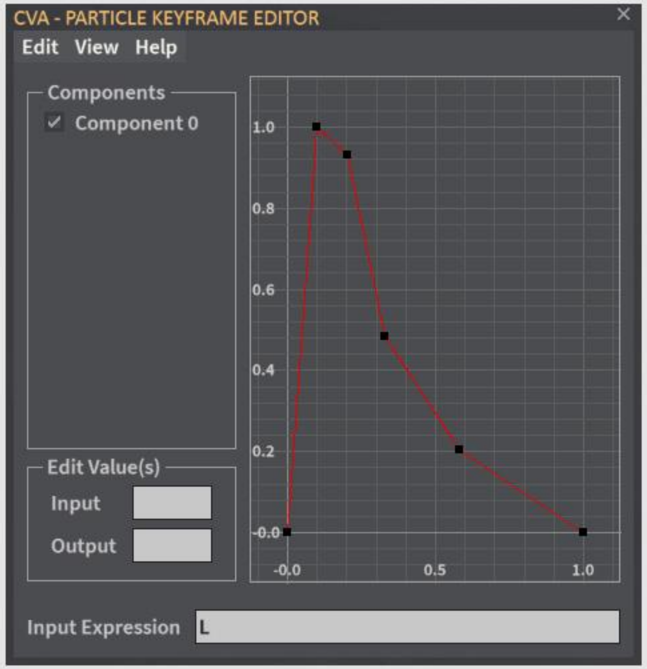
```
▼ fxp_hro_smk_missile_h
    distortion_shockwave_camera
    shockwave
    l_point_flash
    l_point_linger
    straight_streaks
    Flash
    ashflakes
    core_rise_anim_slow
    sparkstreaks_slow
    sparkstreaks_fast
    smoke_sphere
    parent_streamers
    child_streamers_wisps
    child_streamer_glowhead
    child_directional_sparks
    ashflakes_linger
    smoke_sphere_linger
    smoke_ground_ring_linger
```

Outliner



name	child_streamers_wisps
class	PARTICLE_EMITTER_NEW
▼ GENERATION	
emit_parent	parent_streamers
emit_kind	Emit
emit_events	emit(-1, 150)
emit_ratio	
emit_space	kParticleSystem
emit_pos	0 0 0
emit_v	0 0 0
lifetime	parent.childlife * uSmokeLife + parent.plife * abs(1 - parent.L)
coalesce_particles	□
▼ MOTION	
bounce_kind	None
update_space	kWorld
gravity	750 * eval_emit(parent.trailcolor.x)
damping	25
pos	INTEGRATE(POS, V, DV)
v	INTEGRATE(V, DV)

Property Editor



☑ Component 0

Input Expression L

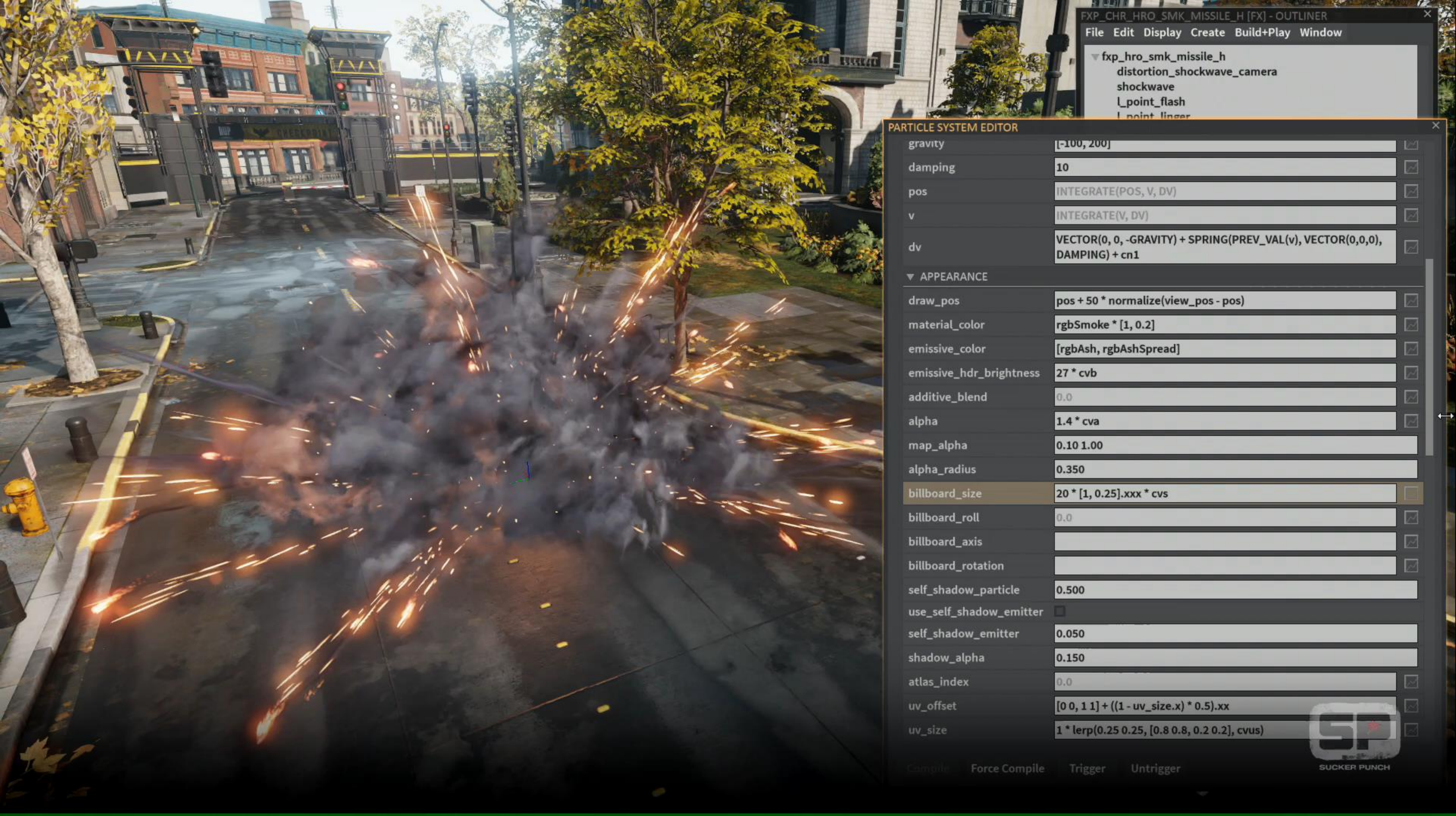
Keyframe Editor

TOOLSET | THE BASICS

- Expressions:
  - Vectors, floats, strings, bools, ints, orientations (quaternion), random ranges
  - Uses standard math operations - cos, abs, dot, swizzling, etc.
  - Once compiled into constants, values can be edited live
- Variables (user parameters) can be created and used in expressions
- Triangulation & blending methods
  - Billboards, Ribbons, Mesh fragments
  - Translucent, Deferred, Additive, Distort
- Diffuse and emissive values + light casting abilities

TOOLSET | EMIT & UPDATE

- Basic emit functions
  - Emit(time, count)
  - Burst(count)
  - Pause(time)
  - Emit Ratio multiplier
- Supports a range of emit and update spaces
  - Local, world, spline, view
  - Unlinked emit and update spaces
  - Can convert and lerp between spaces (positions)



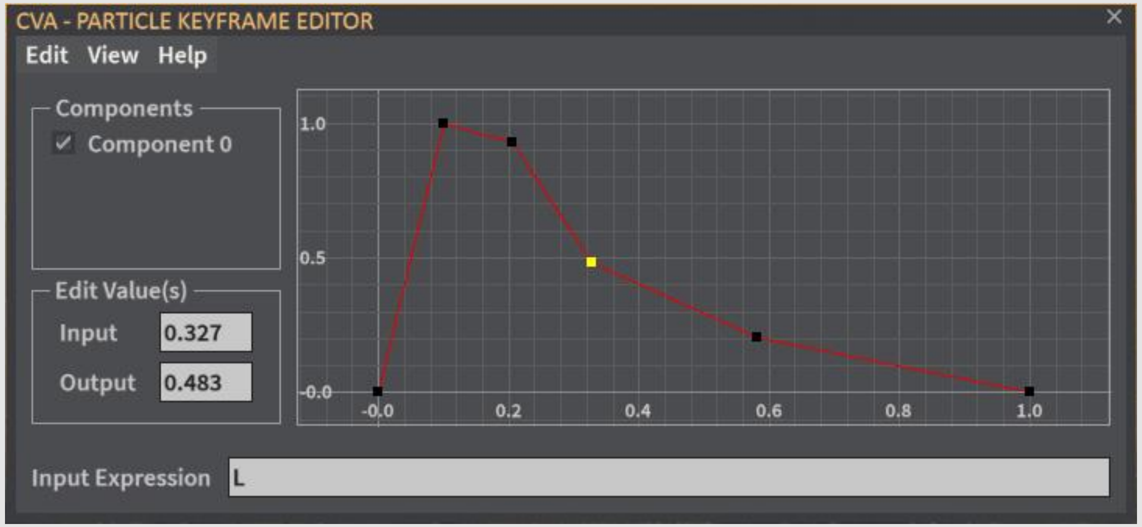
gravity	[-100, 200]
damping	10
pos	INTEGRATE(POS, V, DV)
v	INTEGRATE(V, DV)
dv	VECTOR(0, 0, -GRAVITY) + SPRING(PREV_VAL(v), VECTOR(0,0,0), DAMPING) + cn1
draw_pos	pos + 50 * normalize(view_pos - pos)
material_color	rgbSmoke * [1, 0.2]
emissive_color	[rgbAsh, rgbAshSpread]
emissive_hdr_brightness	27 * cvb
additive_blend	0.0
alpha	1.4 * cva
map_alpha	0.10 1.00
alpha_radius	0.350
billboard_size	20 * [1, 0.25].xxx * cvs
billboard_roll	0.0
billboard_axis	
billboard_rotation	
self_shadow_particle	0.500
use_self_shadow_emitter	□
self_shadow_emitter	0.050
shadow_alpha	0.150
atlas_index	0.0
uv_offset	[0 0, 1 1] + ((1 - uv_size.x) * 0.5).xx
uv_size	1 * lerp(0.25 0.25, [0.8 0.8, 0.2 0.2], cvus)

TOOLSET | USER PARAMETERS

- System level parameters
  - Floats or vectors that can be referenced in any expression on emitters
  - Values can be adjusted externally through script and code
- Emitter level parameters
  - Evaluated per particle such that a random number will be different on each particle per emitter
  - Returns the same value (per particle) when referenced multiple times in the emitter
  - Can evaluate only on emit or return a different value at emit time from subsequent frames

TOOLSET | KEYFRAME EDITOR

- Controls the output of floats, vec2 and vec3
- Linear segments, limited to 16 points
- The x-axis input on the graph can be set to other parameters for added power and flexibility



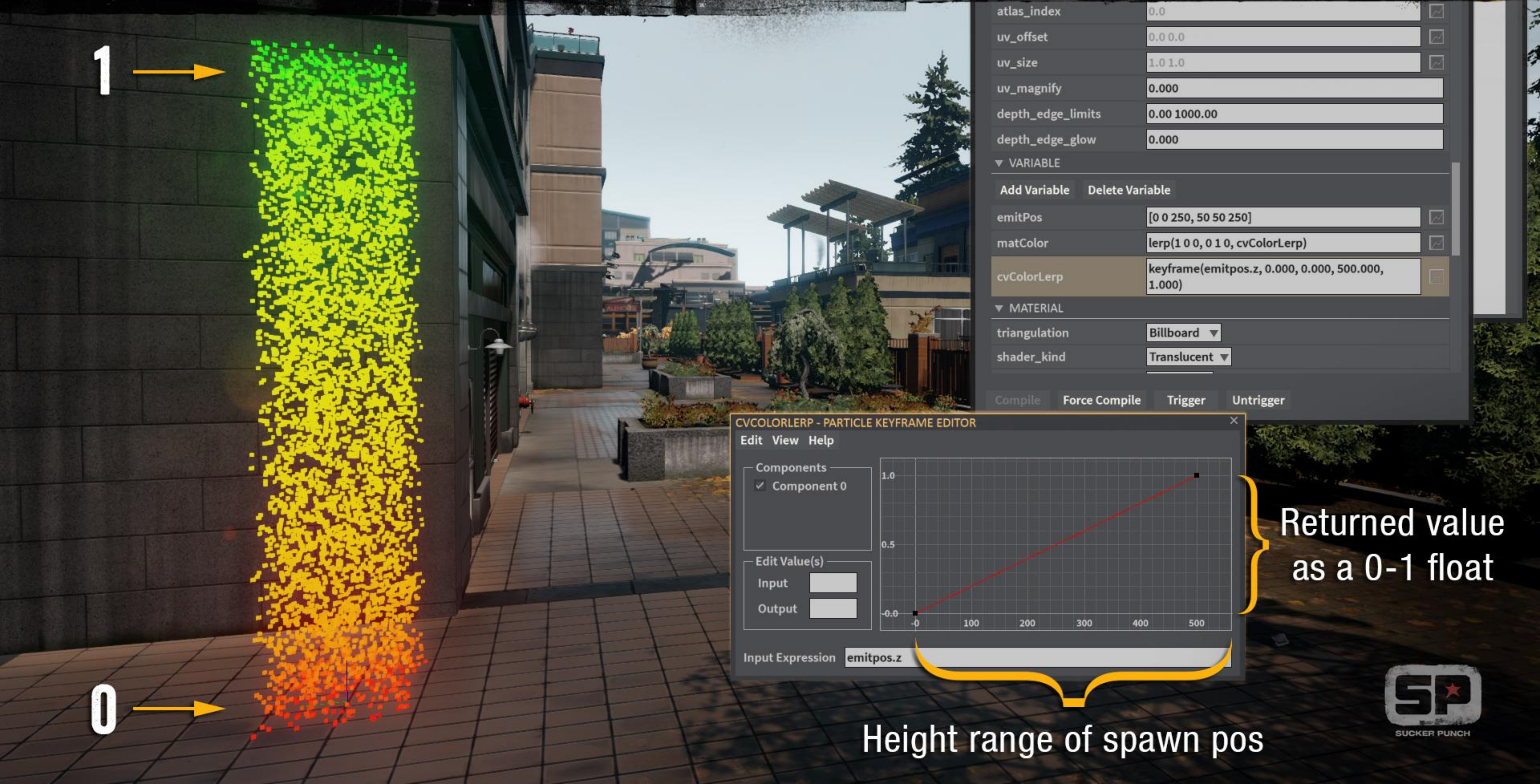
☑ Component 0

Input 0.327

Output 0.483

Input Expression L

TOOLSET | KEYFRAME EDITOR



atlas_index	0.0
uv_offset	0.0 0.0
uv_size	1.0 1.0
uv_magnify	0.000
depth_edge_limits	0.00 1000.00
depth_edge_glow	0.000

▼ VARIABLE

emitPos	[0 0 250, 50 50 250]
matColor	lerp(1 0 0, 0 1 0, cvColorLerp)
cvColorLerp	keyframe(emitpos.z, 0.000, 0.000, 500.000, 1.000)

▼ MATERIAL

triangulation	Billboard ▼
shader_kind	Translucent ▼

Components

☑ Component 0

Edit Value(s)

Input

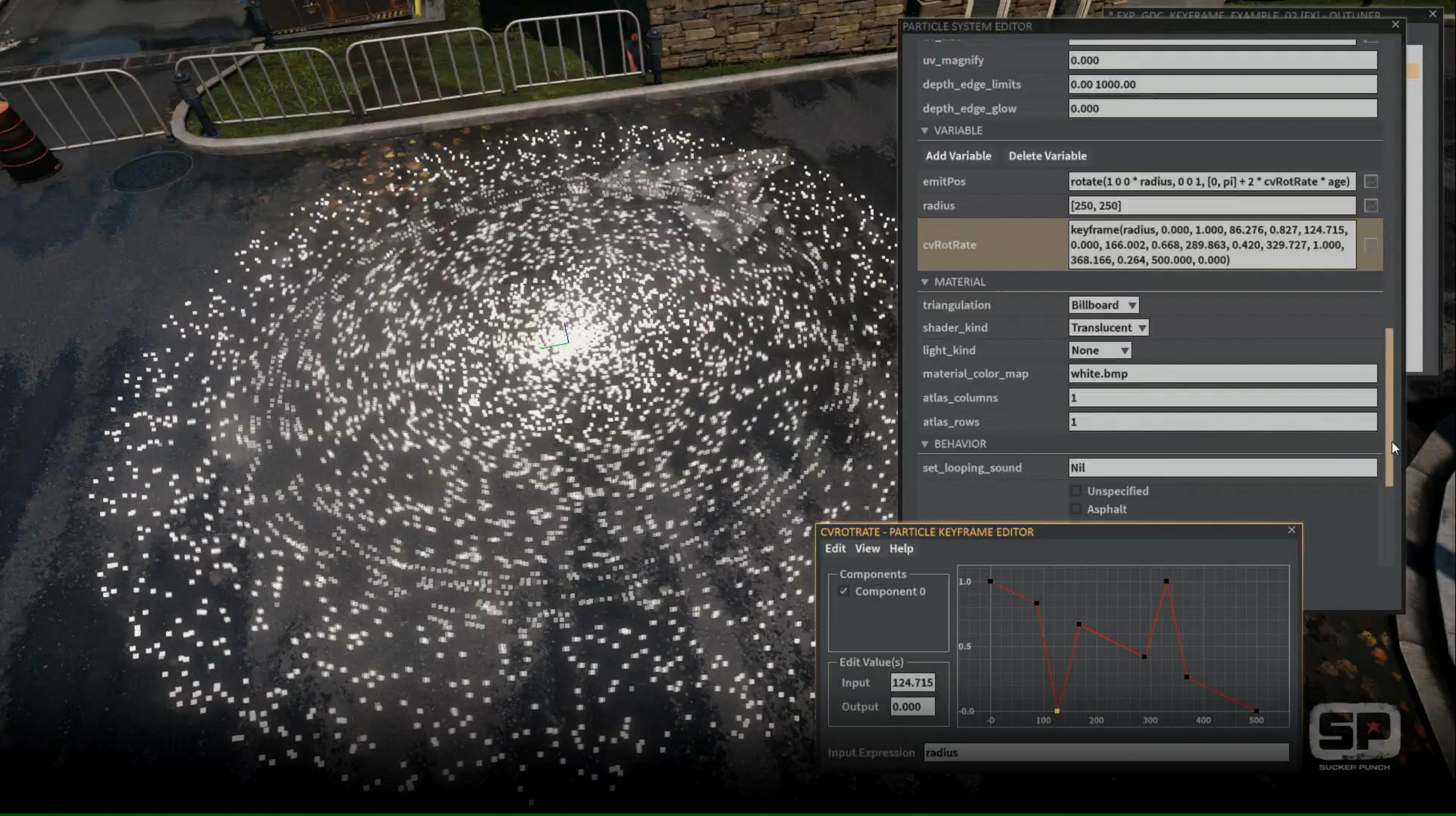
Output

Input Expression: emitpos.z

Returned value as a 0-1 float

Height range of spawn pos

SUCKER PUNCH



uv_magnify 0.000

depth_edge_limits 0.00 1000.00

depth_edge_glow 0.000

▼ VARIABLE

emitPos rotate(1 0 0 * radius, 0 0 1, [0, pi] + 2 * cvRotRate * age)

radius [250, 250]

cvRotRate keyframe(radius, 0.000, 1.000, 86.276, 0.827, 124.715, 0.000, 166.002, 0.668, 289.863, 0.420, 329.727, 1.000, 368.166, 0.264, 500.000, 0.000)

▼ MATERIAL

triangulation Billboard

shader_kind Translucent

light_kind None

material_color_map white.bmp

atlas_columns 1

atlas_rows 1

▼ BEHAVIOR

set_looping_sound Nil

□ Unspecified

□ Asphalt

Components

✓ Component 0

Edit Value(s)

Input 124.715

Output 0.000

Input Expression radius

SHADING & LIGHTING

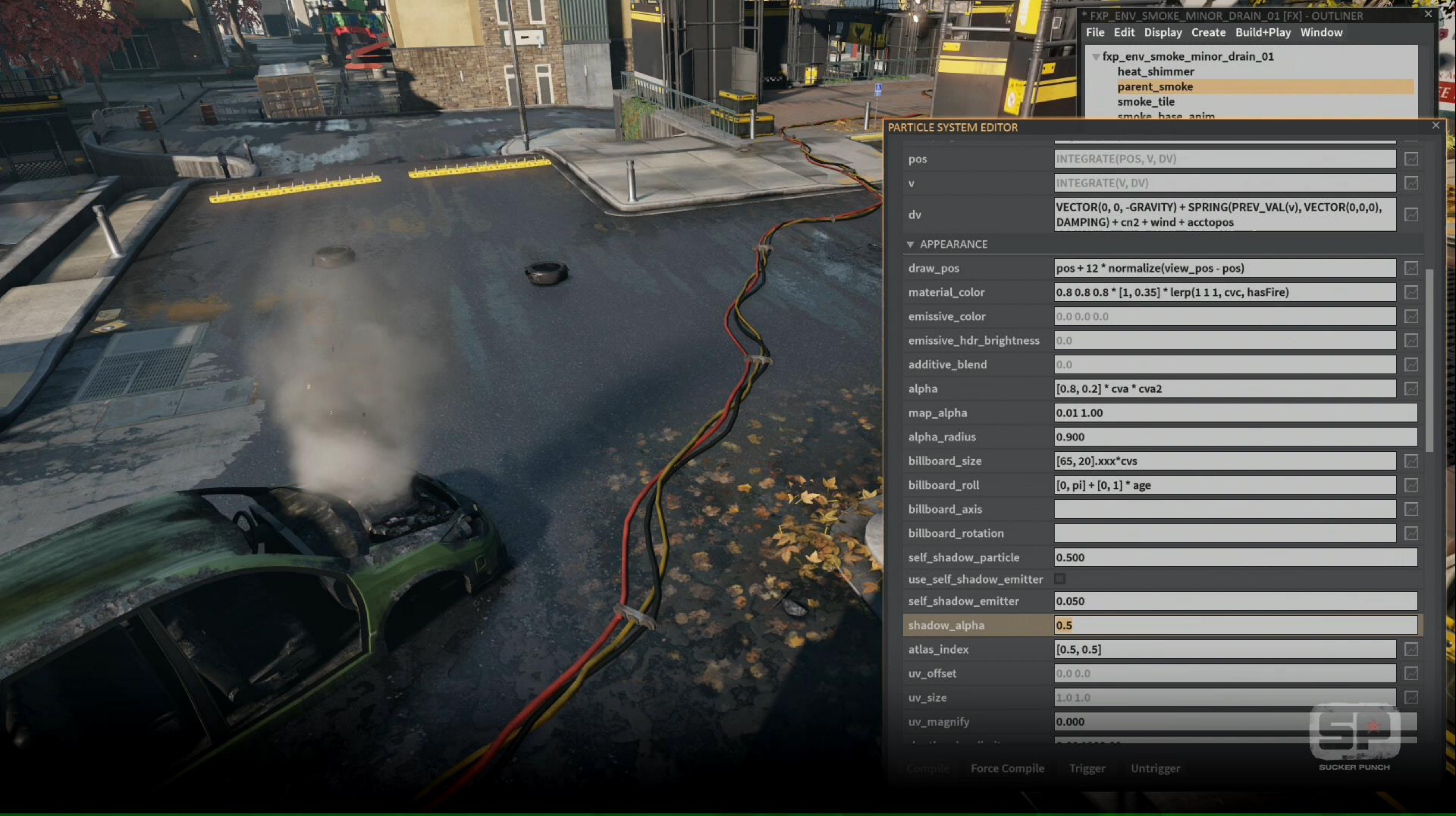
CASE STUDY | SHADING & LIGHTING

- Cast shadows
- Receive shadows
- Cast lights
- Bounced ambient
- Receive directional sunlight
- Blend correctly with haze
- HDR particle rendering

SUCKER PUNCH

SHADING & LIGHTING | SHADOWS

- Simple multiplicative blob shadows, one per particle
- Tunable shadow strength per emitter
- Deferred rendered geometry meshes cast shadows like all world geometry
- Additive blended particles do not cast shadows
- Receive shadows cast from geometry affected by directional lighting
- Both dynamic and static objects cast shadows onto particles



- fxp_env_smoke_minor_drain_01
  - heat_shimmer
  - parent_smoke
  - smoke_tile
  - smoke_base_anim

pos	INTEGRATE(POS, V, DV)
v	INTEGRATE(V, DV)
dv	VECTOR(0, 0, -GRAVITY) + SPRING(PREV_VAL(v), VECTOR(0,0,0), DAMPING) + cn2 + wind + acctopos

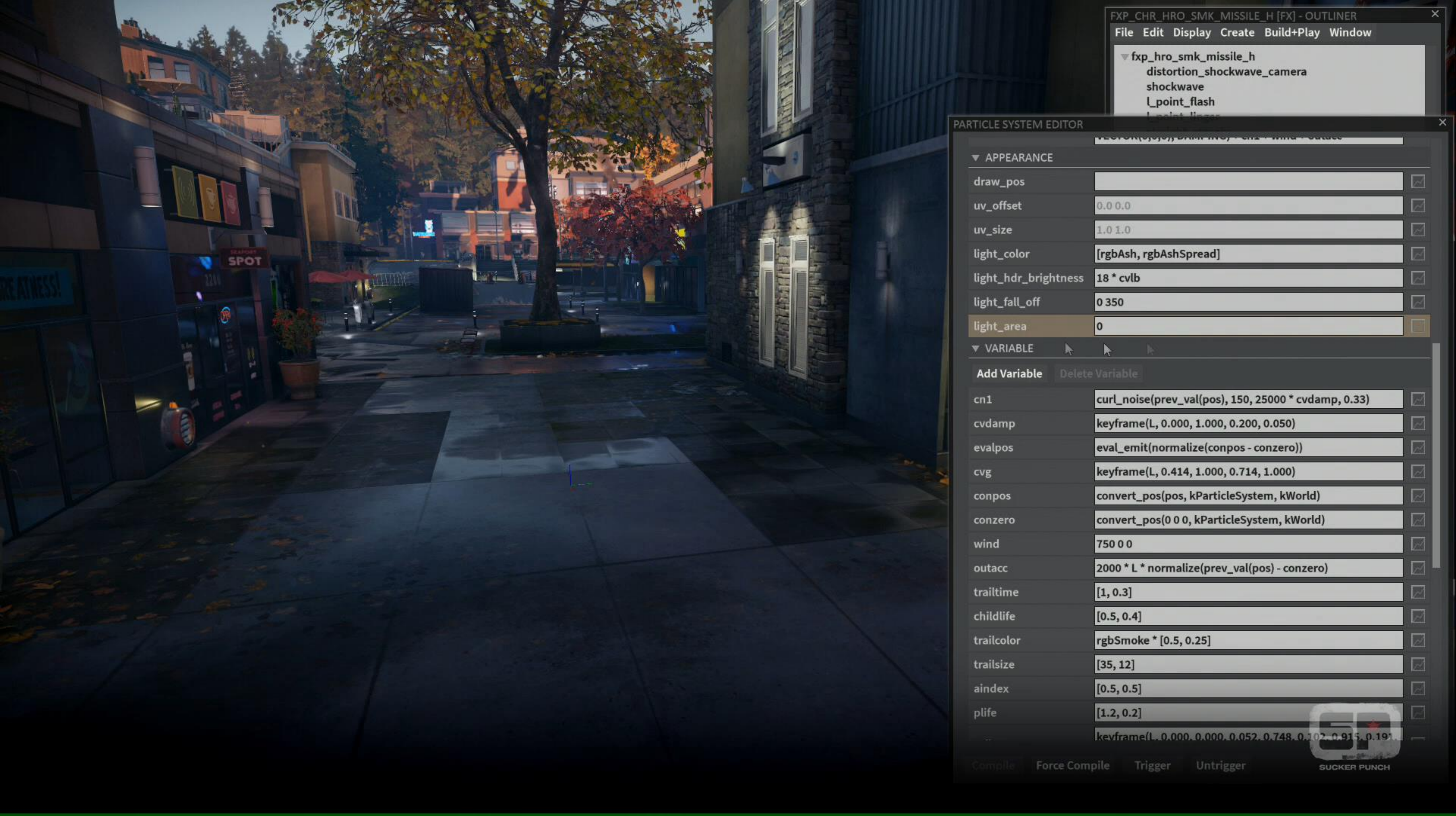
▼ APPEARANCE

draw_pos	pos + 12 * normalize(view_pos - pos)
material_color	0.8 0.8 0.8 * [1, 0.35] * lerp(1 1 1, cvc, hasFire)
emissive_color	0.0 0.0 0.0
emissive_hdr_brightness	0.0
additive_blend	0.0
alpha	[0.8, 0.2] * cva * cva2
map_alpha	0.01 1.00
alpha_radius	0.900
billboard_size	[65, 20].xxx*cvs
billboard_roll	[0, pi] + [0, 1] * age
billboard_axis	
billboard_rotation	
self_shadow_particle	0.500
use_self_shadow_emitter	□
self_shadow_emitter	0.050
shadow_alpha	0.5
atlas_index	[0.5, 0.5]
uv_offset	0.0 0.0
uv_size	1.0 1.0
uv_magnify	0.000

SHADING & LIGHTING | CAST LIGHTS

- Can cast point lights from any particle
- Falloff & hotspot controls
- HDR values
- All the same positional expressions as a particle
- Does not affect translucent particles

SUCKER PUNCH



VECTOR(0,0,0), DAMPING) + cn1 + wind + outacc

▼ APPEARANCE

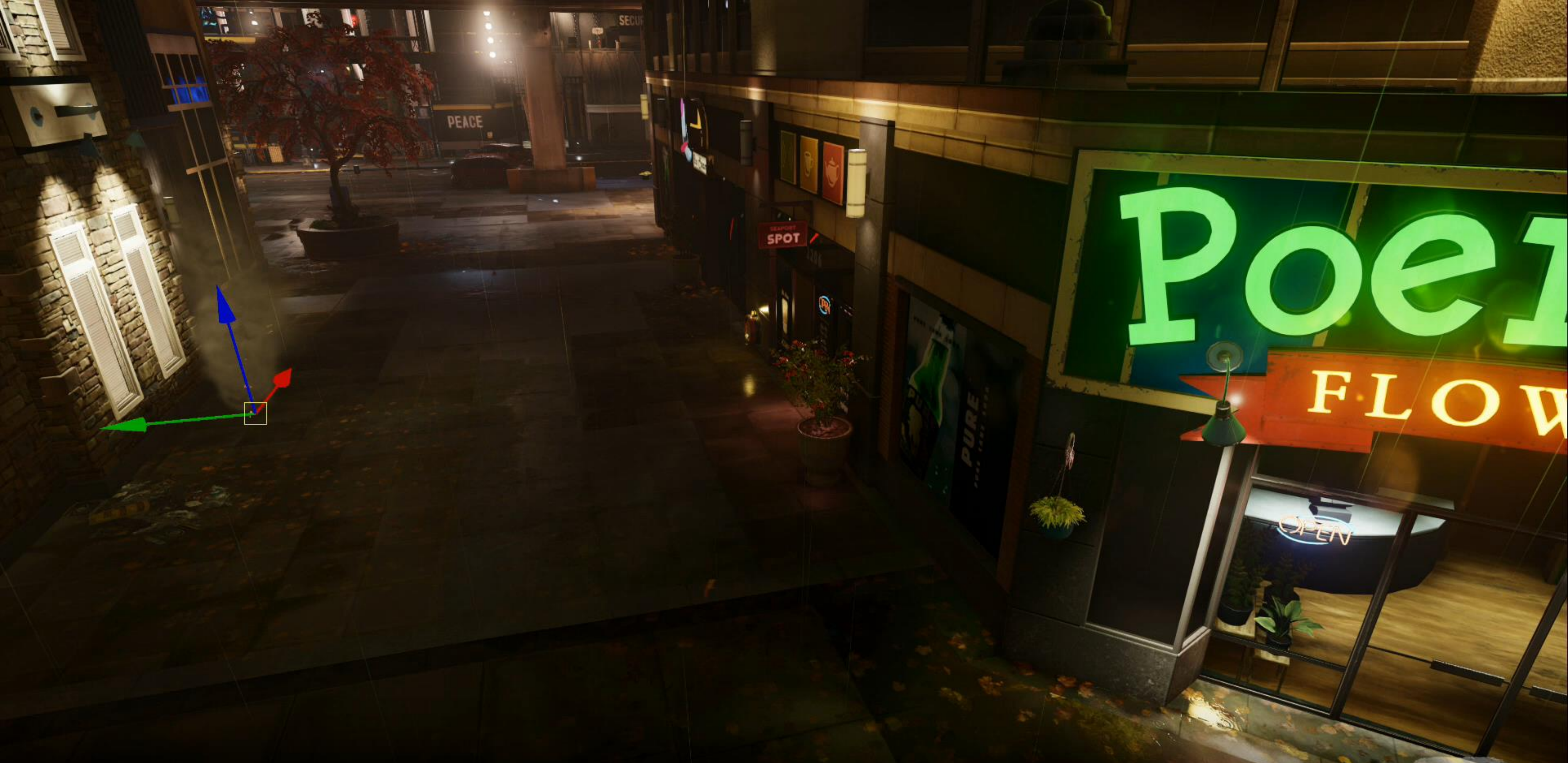
draw_pos	
uv_offset	0.0 0.0
uv_size	1.0 1.0
light_color	[rgbAsh, rgbAshSpread]
light_hdr_brightness	18 * cvlb
light_fall_off	0 350
light_area	0

▼ VARIABLE

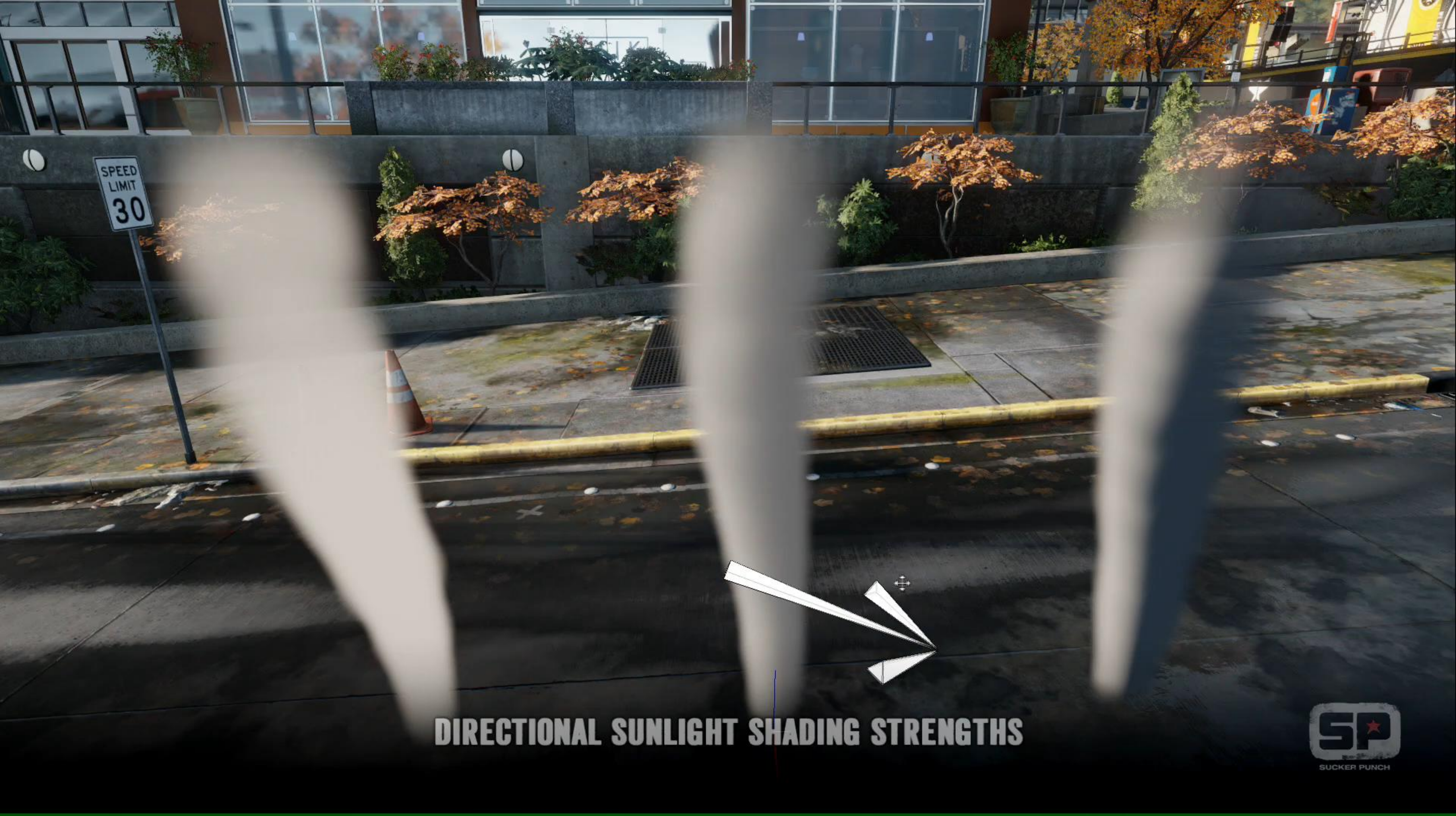
cn1	curl_noise(prev_val(pos), 150, 25000 * cvdamp, 0.33)
cvdamp	keyframe(L, 0.000, 1.000, 0.200, 0.050)
evalpos	eval_emit(normalize(conpos - conzero))
cvg	keyframe(L, 0.414, 1.000, 0.714, 1.000)
conpos	convert_pos(pos, kParticleSystem, kWorld)
conzero	convert_pos(0 0 0, kParticleSystem, kWorld)
wind	750 0 0
outacc	2000 * L * normalize(prev_val(pos) - conzero)
trailtime	[1, 0.3]
childlife	[0.5, 0.4]
trailcolor	rgbSmoke * [0.5, 0.25]
trailsize	[35, 12]
aindex	[0.5, 0.5]
plife	[1.2, 0.2]
	keyframe(L, 0.000, 0.000, 0.052, 0.748, 0.102, 0.915, 0.19

SHADING & LIGHTING | AMBIENT SH PROBE LIGHTING

- Spherical Harmonics probe data used on every (non-additive) particle emitter
- Simulates local and bounced lighting
- Huge success in shadowed areas which would ordinarily look flat from lack of directional sunlight



AMBIENT SH PROBE LIGHTING



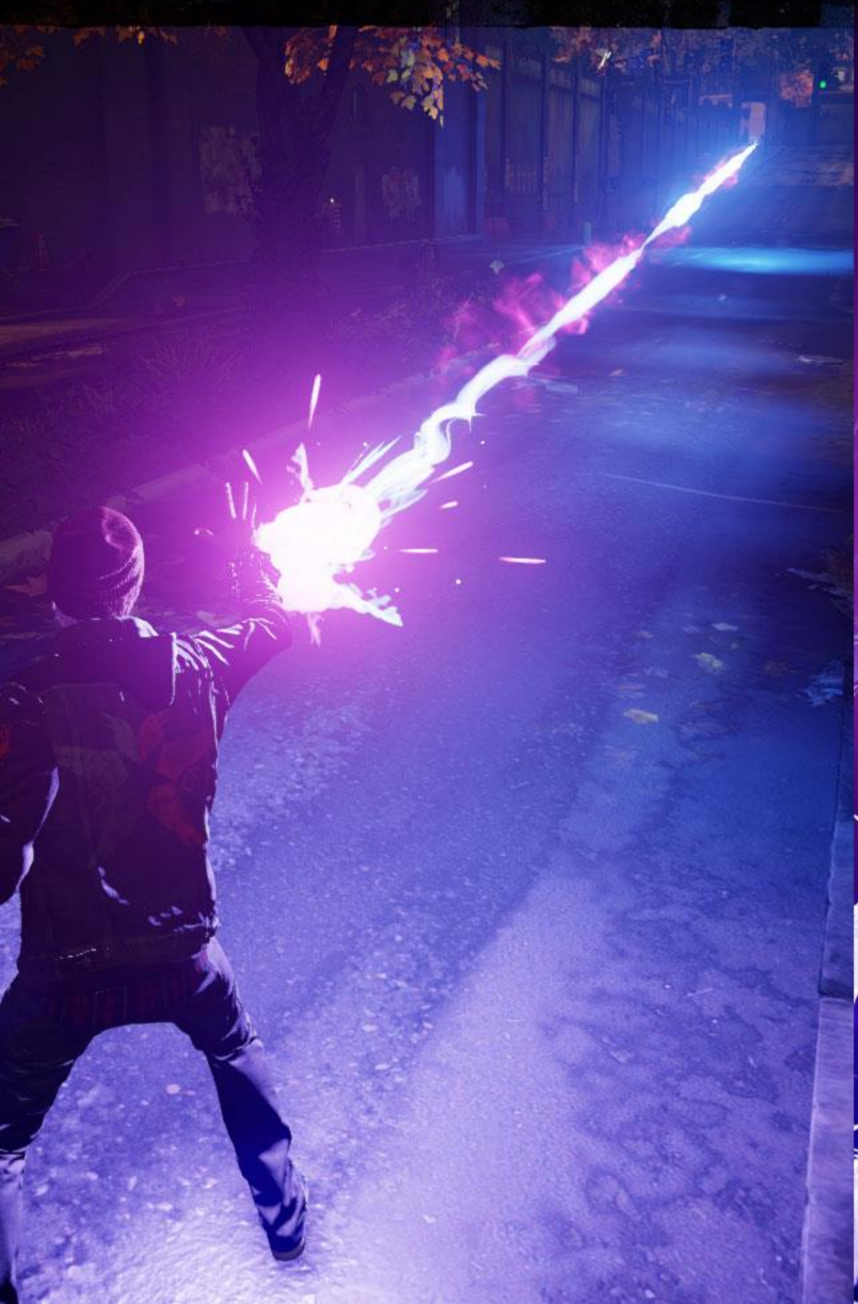
DIRECTIONAL SUNLIGHT SHADING STRENGTHS

SUCKER PUNCH

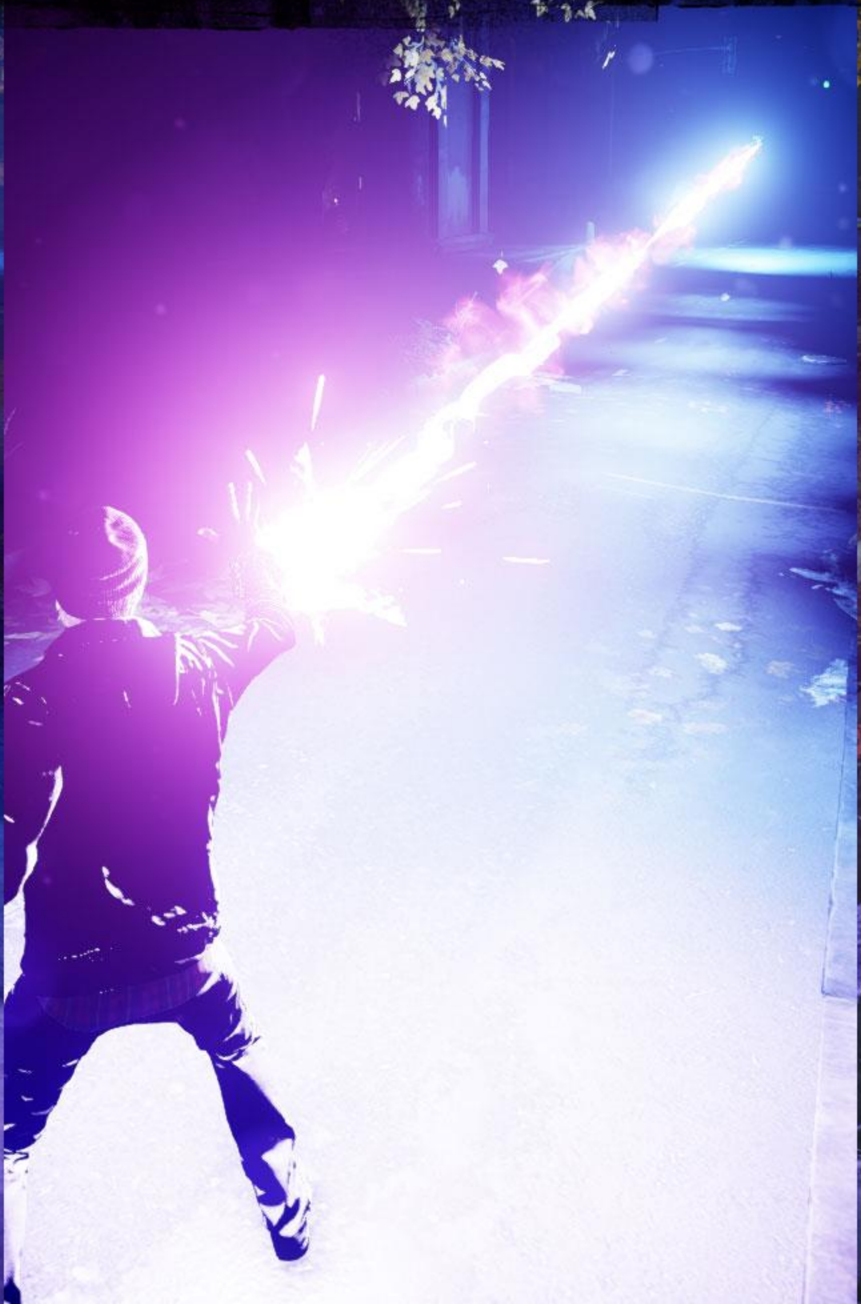
SHADING & LIGHTING | HDR PIPELINE

- I:SS uses Physically Based Rendering / HDR
- Started out using realistic exposure values for emitters but altered to suit artistic and design needs
- HDR offset per time of day used to compensate
    - 8 different times of day
- Particle textures authored in LDR, very easy to ‘blow out’ the alpha

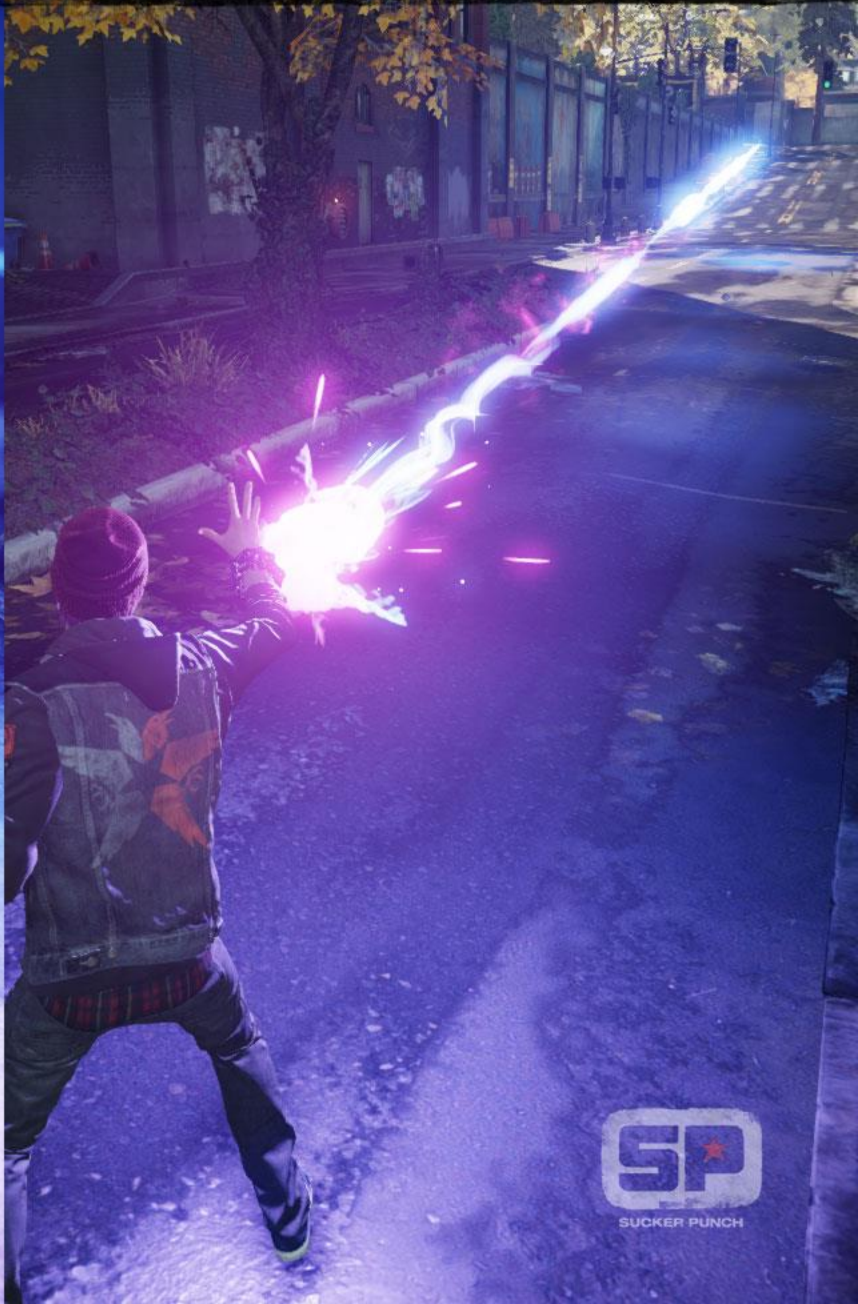
SUNSET

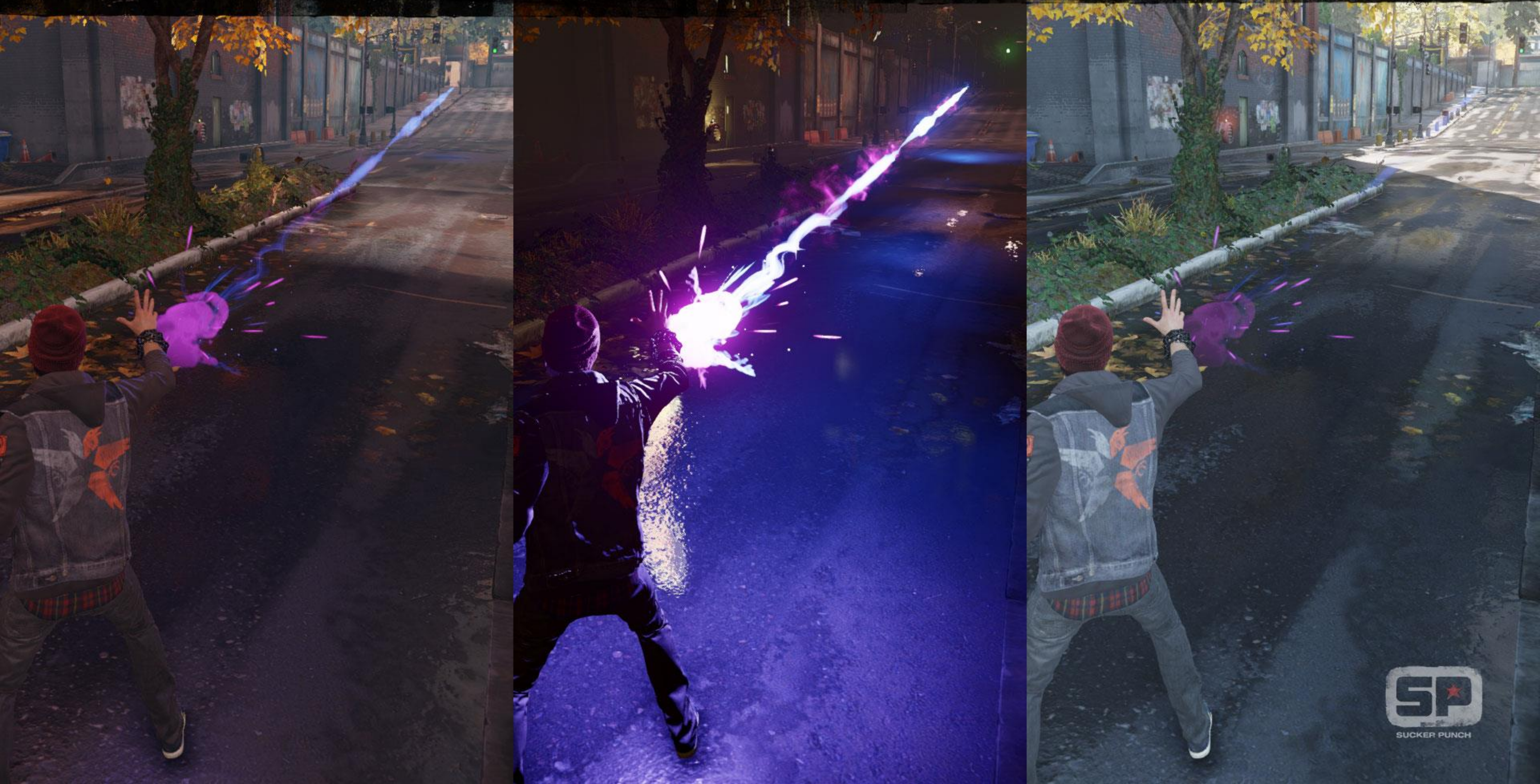


NIGHT



DAY

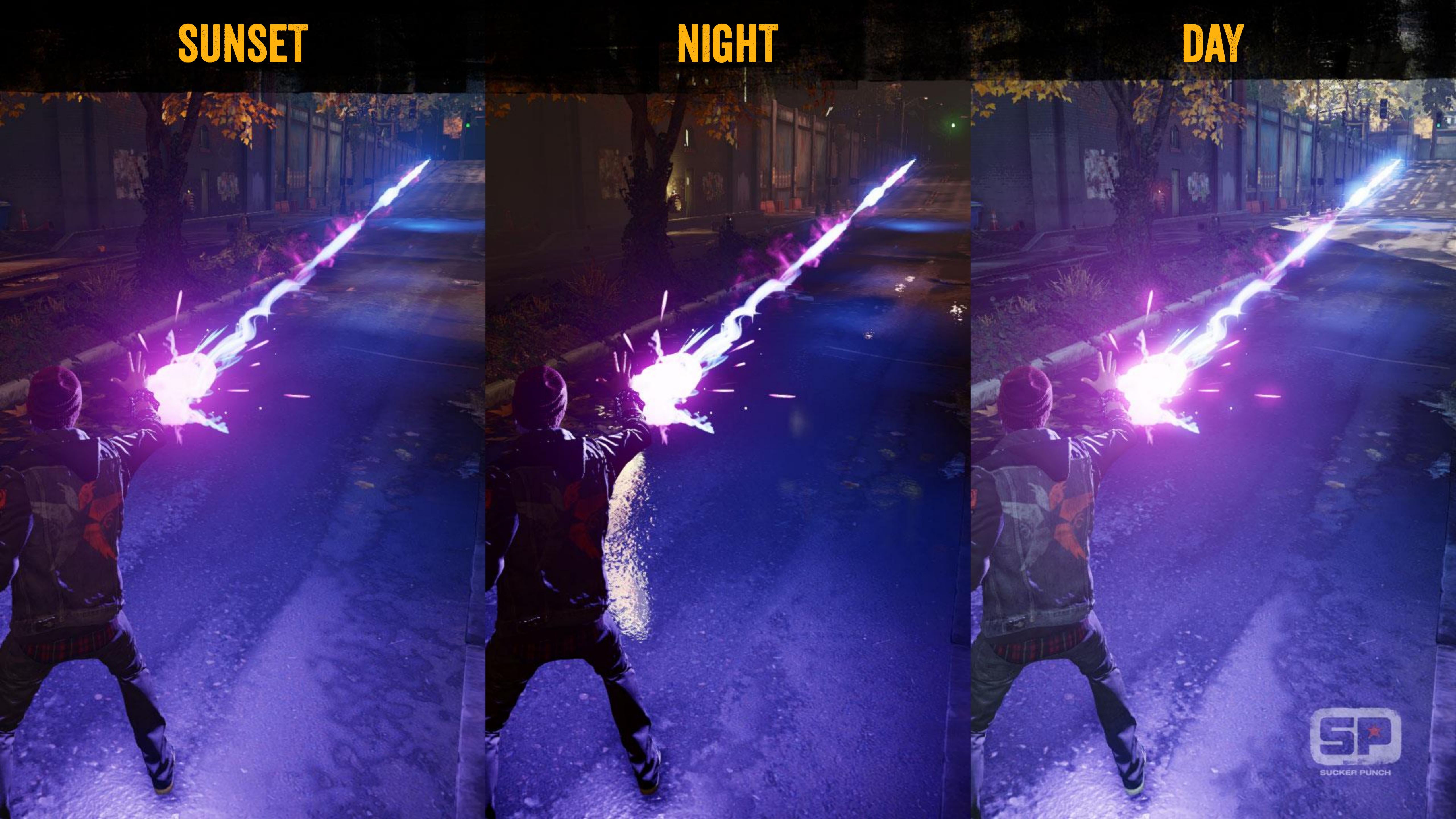




SUNSET

NIGHT

DAY



SUNSET

NIGHT

DAY

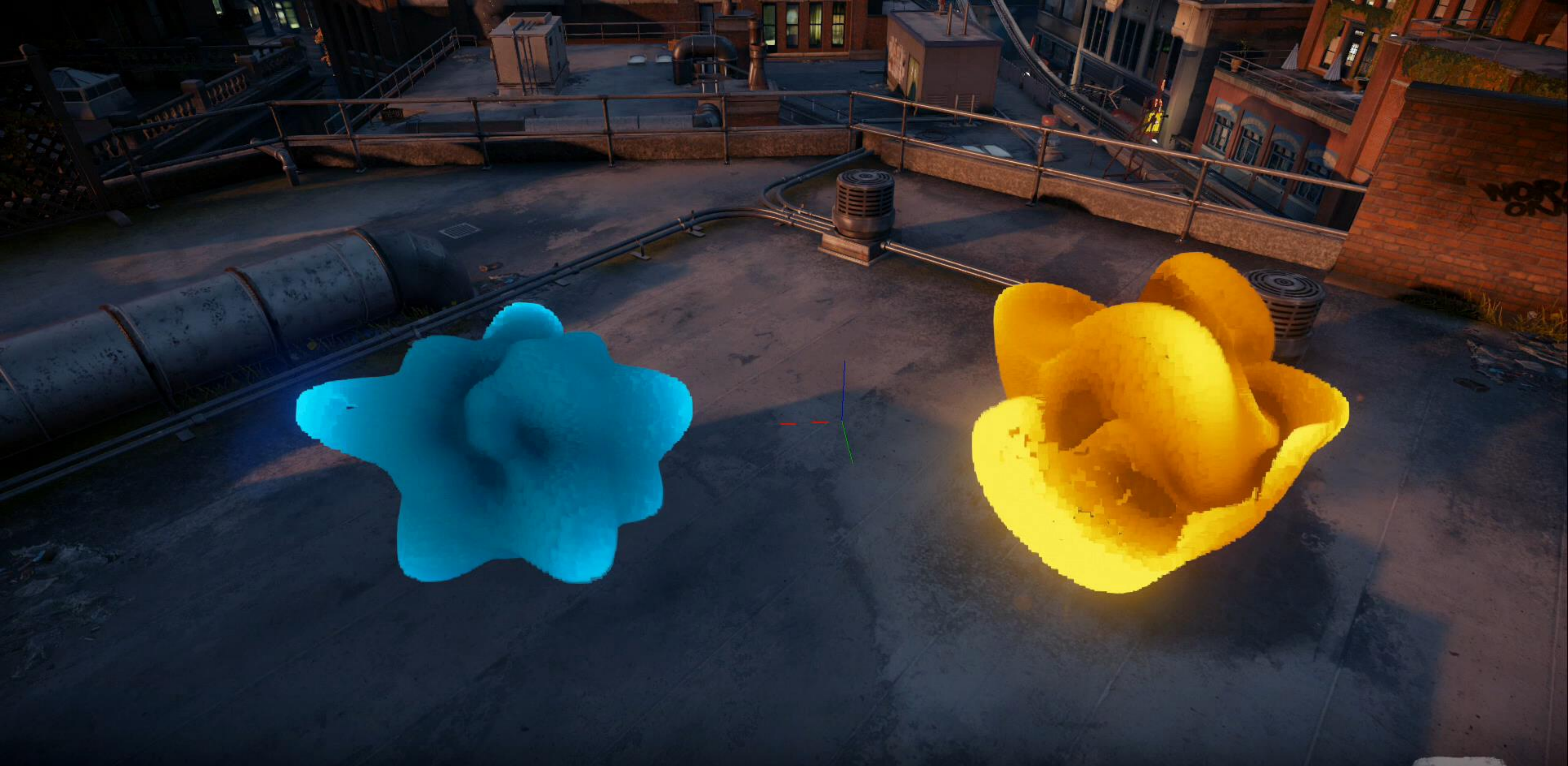
CURL NOISE

CURL NOISE | BASICS

- 1D Simplex noise (variant of Perlin)
  - Returns a float
- 3D noise (built from 1D noise + Bill Rockenbeck magic)
  - Returns a vector
- Contains an input position, frequency, strength & iteration time
  - curl_noise(pos, frequency, strength, dt)

DAMPED CURL

UN-DAMPED CURL



BLUE: 1D RADIUS NOISE | ORANGE: 3D POSITIONAL NOISE

POST-MORTEM

SUCKER PUNCH

POST-MORTEM | PROS

- Nearly limitless power over simulations, the only limit is our understanding of math
- Easy to have engineers prototype functionality that can be switched to code or simplified expression functions later
- Very flexible and unspecialized system
- GPU particles are fast so we can have very large quantities
- Real-time editing a huge boon to iteration time
- Great particle sorting and lighting

POST-MORTEM | PROS

- Very flexible and unspecialized system, the only limit is our understanding of math
- Easy to have engineers prototype functionality that can be switched to code or simplified expression functions later
- GPU particles are fast so we can have very large quantities
- Real-time editing a huge boon to iteration time
- Great particle sorting and lighting

POST-MORTEM | PROS

- Very flexible and unspecialized system, the only limit is our understanding of math
- Easy to have engineers prototype functionality that can be switched to code or simplified expression functions later
- GPU particles are fast so we can have very large quantities
- Real-time editing a huge boon to iteration time
- Great particle sorting and lighting

POST-MORTEM | PROS

- Very flexible and unspecialized system, the only limit is our understanding of math
- Easy to have engineers prototype functionality that can be switched to code or simplified expression functions later
- GPU particles are fast so we can have very large quantities
- Real-time editing a huge boon to iteration time
- Great particle sorting and lighting

POST-MORTEM | PROS

- Very flexible and unspecialized system, the only limit is our understanding of math
- Easy to have engineers prototype functionality that can be switched to code or simplified expression functions later
- GPU particles are fast so we can have very large quantities
- Real-time editing a huge boon to iteration time
- Great particle sorting and lighting

SUCKER PUNCH

POST-MORTEM | PROS

- Very flexible and unspecialized system, the only limit is our understanding of math
- Easy to have engineers prototype functionality that can be switched to code or simplified expression functions later
- GPU particles are fast so we can have very large quantities
- Real-time editing a huge boon to iteration time
- Great particle sorting and lighting

POST-MORTEM | CONS

- Systems can get extremely complex very quickly
- It can be challenging to work on another artists systems
- Very limited shading options, lack of shader editing
- Not artist-friendly, which makes finding the right effects artists and training them much more difficult
- Our HDR lighting model is much more challenging to work with given different time of day and lighting scenarios

POST-MORTEM | CONS

- Systems can get extremely complex very quickly
- It can be challenging to work on another artists systems
- Very limited shading options, lack of shader editing
- Not artist-friendly, which makes finding the right effects artists and training them much more difficult
- Our HDR lighting model is much more challenging to work with given different time of day and lighting scenarios

POST-MORTEM | CONS

- Systems can get extremely complex very quickly
- It can be challenging to work on another artists systems
- Very limited shading options, lack of shader editing
- Not artist-friendly, which makes finding the right effects artists and training them much more difficult
- Our HDR lighting model is much more challenging to work with given different time of day and lighting scenarios

POST-MORTEM | CONS

- Systems can get extremely complex very quickly
- It can be challenging to work on another artists systems
- Very limited shading options, lack of shader editing
- Not artist-friendly, which makes finding the right effects artists and training them much more difficult
- Our HDR lighting model is much more challenging to work with given different time of day and lighting scenarios

POST-MORTEM | CONS

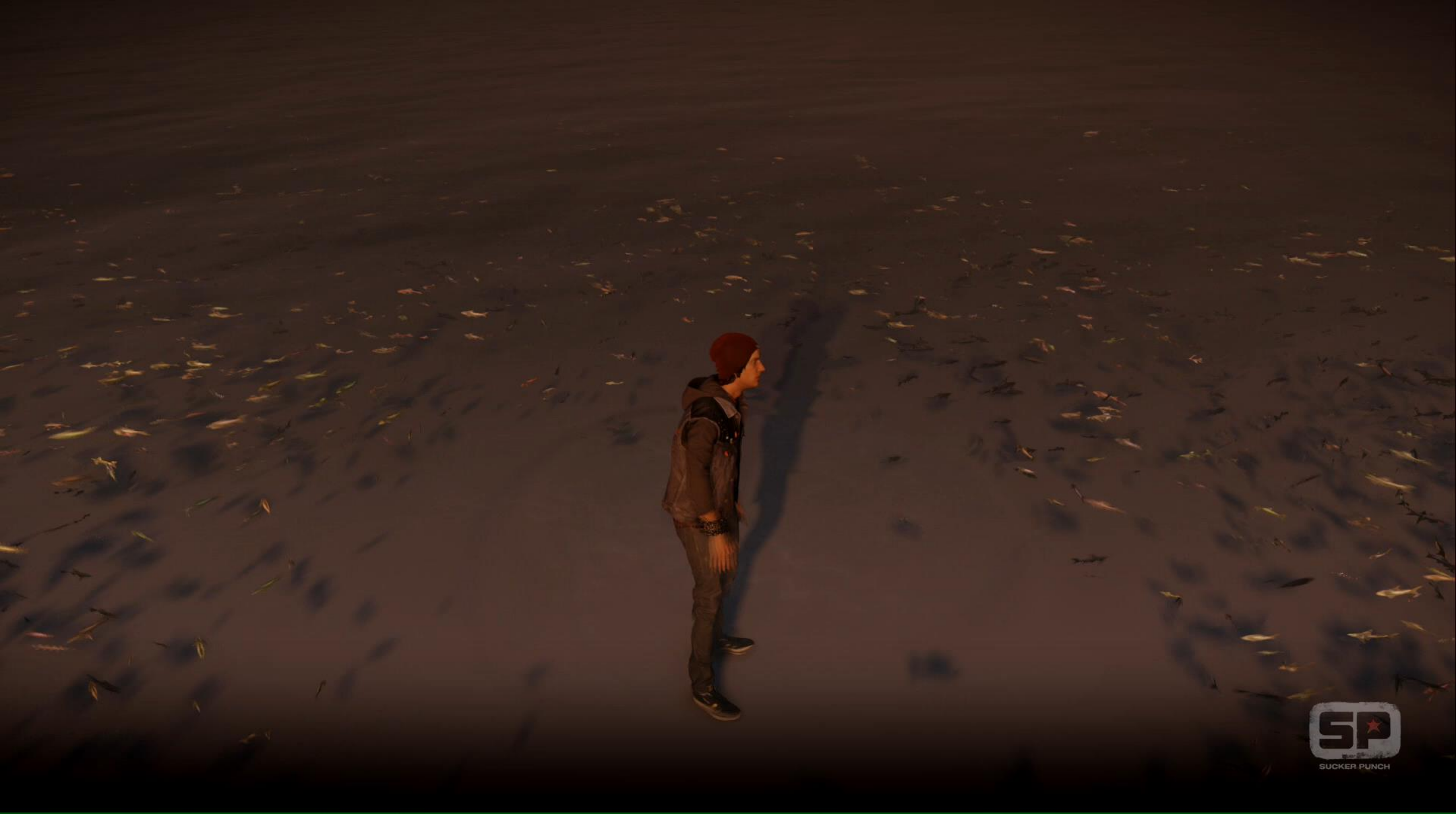
- Systems can get extremely complex very quickly
- It can be challenging to work on another artists systems
- Very limited shading options, lack of shader editing
- Not artist-friendly, which makes finding the right effects artists and training them much more difficult
- Our HDR lighting model is much more challenging to work with given different time of day and lighting scenarios

POST-MORTEM | CONS

- Systems can get extremely complex very quickly
- It can be challenging to work on another artists systems
- Very limited shading options, lack of shader editing
- Not artist-friendly, which makes finding the right effects artists and training them much more difficult
- Our HDR lighting model is much more challenging to work with given different time of day and lighting scenarios

POST-MORTEM | DOING SOMETHING NEW

- If you know how to do it, that means people have already seen it
- Use first principles, don’t pre-constrain your options
- “What do we need to communicate to the player here?”



SP
SUCKER PUNCH

THE END!

QUESTIONS?

mattv@suckerpunch.com

COME VISIT US!

www.suckerpunch.com

soniaj@suckerpunch.com

THE INFAMOUS: SECOND SON PARTICLE SYSTEM ARCHITECTURE

Bill Rockenbeck

Room 2020, West Hall

Friday, March 21

10:00am-11:00am

SUCKER PUNCH