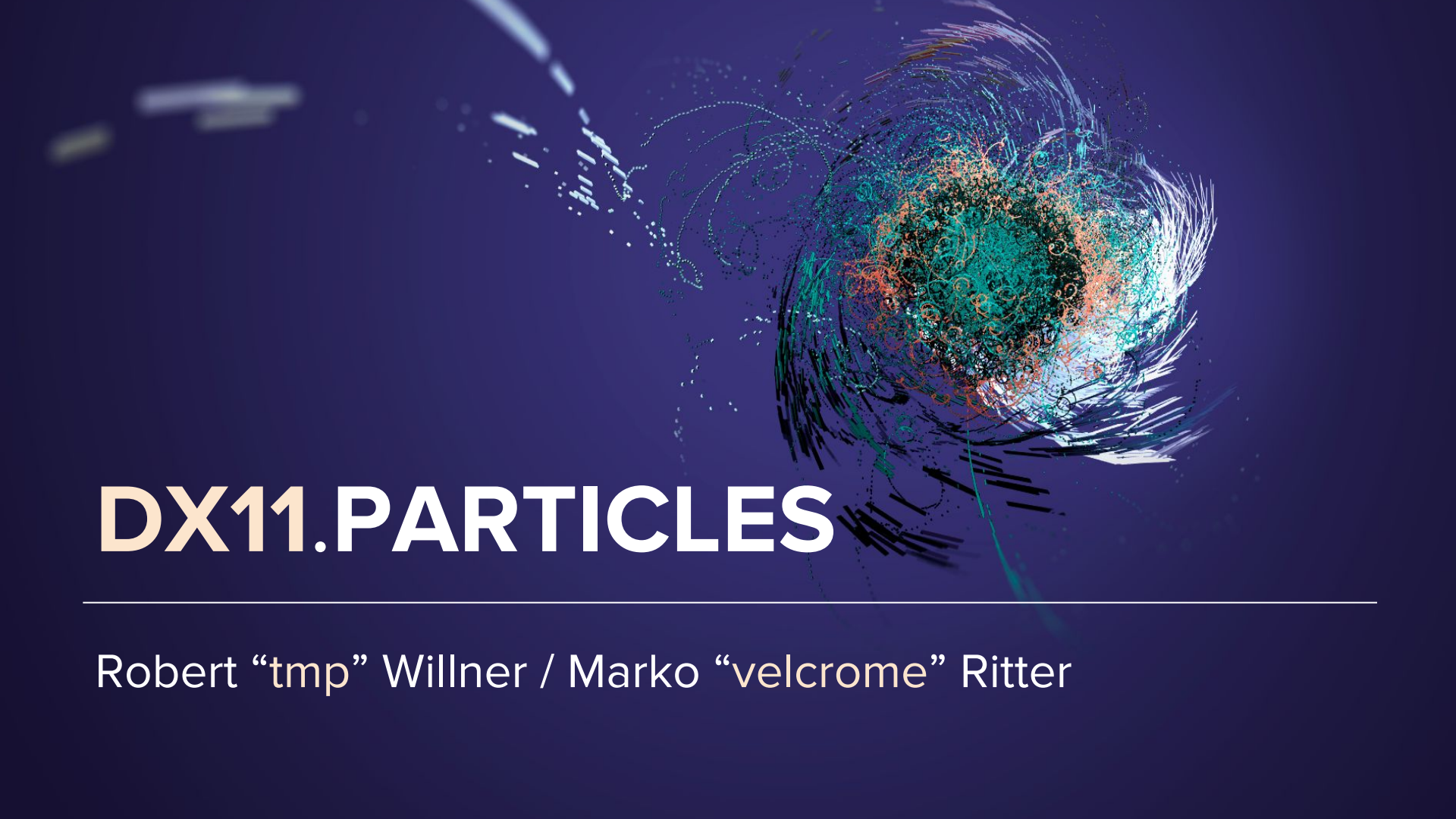


DX11.PARTICLES

Robert “tmp” Willner / Marko “velcrome” Ritter

PREREQUISITES

- VVVV + Addons (x86 or x64)
- DX11 Nodes (x86 or x64)
- DX11.Particles (x86 or x64)
- Instance Noodles

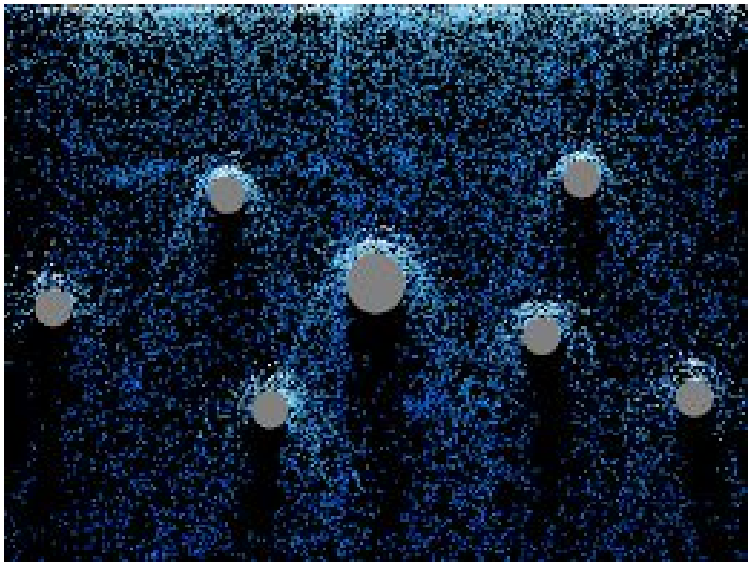
CONTENT

I. Short introduction to particle systems in general

II. Patching Time
   A. Basic Patch (Emitters, Modifiers, Selectors)
   B. Optical Flow Fun (emit + apply forces from moving parts in video input)
   C. Instanced Geometry Noodles (emit from geometry + adaption to InstanceNoodles)
   D. Shooting Range (creative patching + test patches with VIVE)

I. SHORT INTRODUCTION TO PARTICLE SYSTEMS IN GENERAL

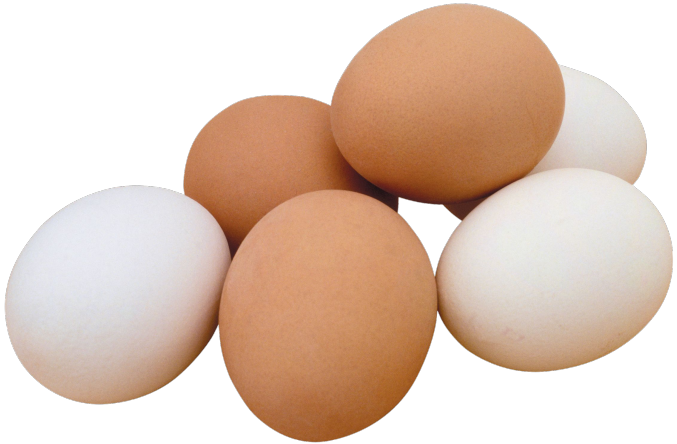
a **particlesystem** is a technique in computer graphics that uses a large number of graphic objects to simulate processes like fire, smoke, water, leaves, ...



I. SHORT INTRODUCTION TO PARTICLE SYSTEMS IN GENERAL

a **particle** can have many parameters:

- lifespan
- position
- force / velocity
- size
- color
- rotation
- ...



I. SHORT INTRODUCTION TO PARTICLE SYSTEMS IN GENERAL

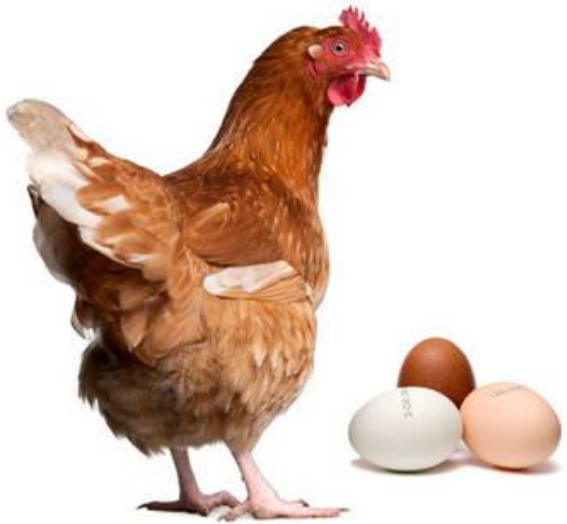
a typical particle systems **update loop** can be separated in **two stages**:

1) **EMISSION** + **SIMULATION** STAGE
2) **RENDERING** STAGE

I. SHORT INTRODUCTION TO PARTICLE SYSTEMS IN GENERAL

1) **EMISSION** + SIMULATION STAGE

- spawning rate
- position
- lifespan



I. SHORT INTRODUCTION TO PARTICLE SYSTEMS IN GENERAL

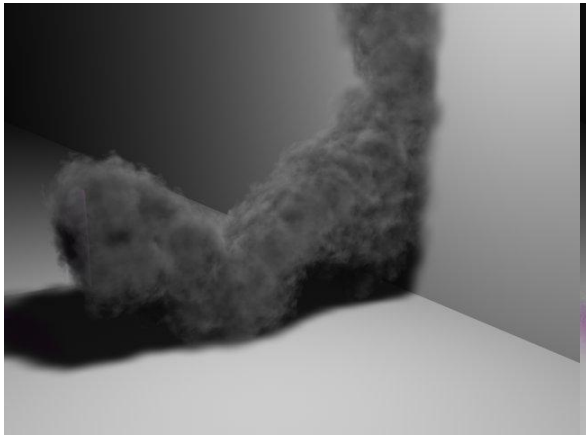
1) EMISSION + **SIMULATION** STAGE

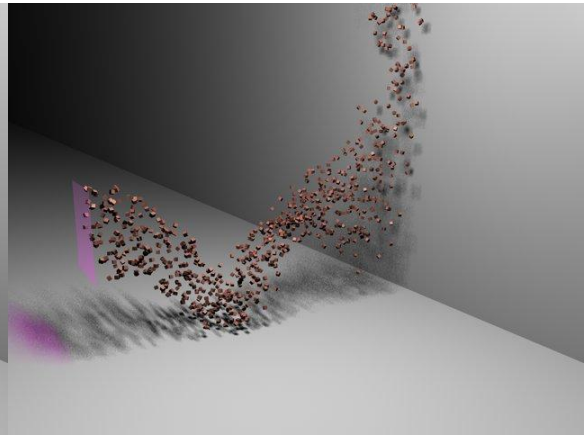
- check if particles have exceeded their lifetime
- parameters are updated

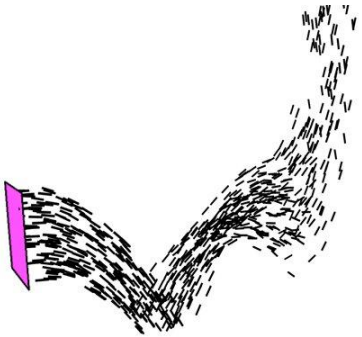
I. SHORT INTRODUCTION TO PARTICLE SYSTEMS IN GENERAL

2) **RENDERING** STAGE

- point
- sprite
- geometry

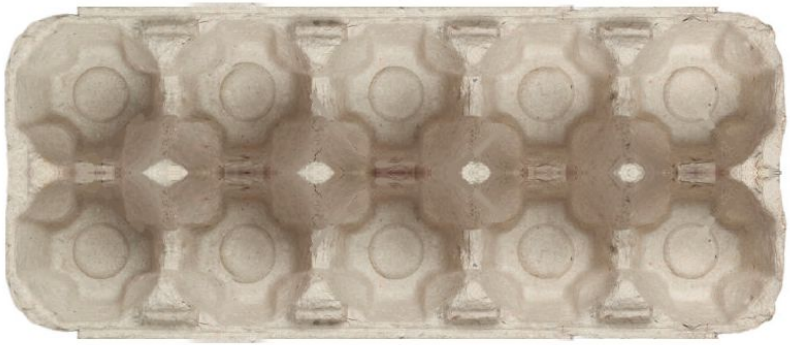






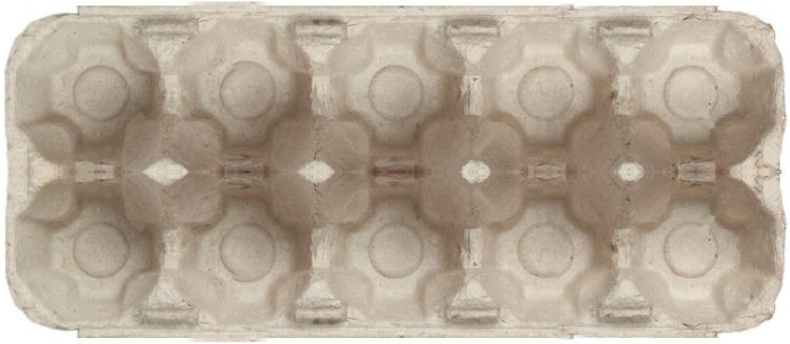
PATCHING TIME

EMITTERS



Emitter Size = 10

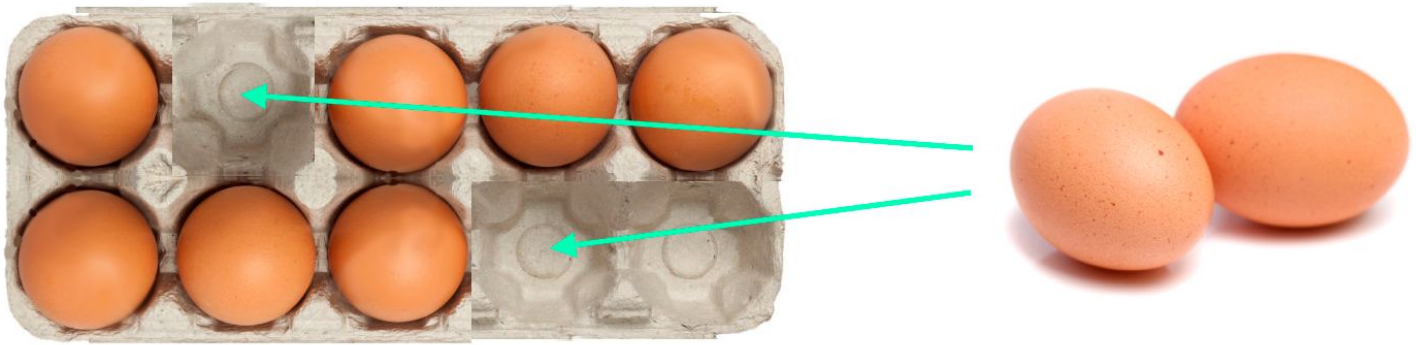
EMITTERS



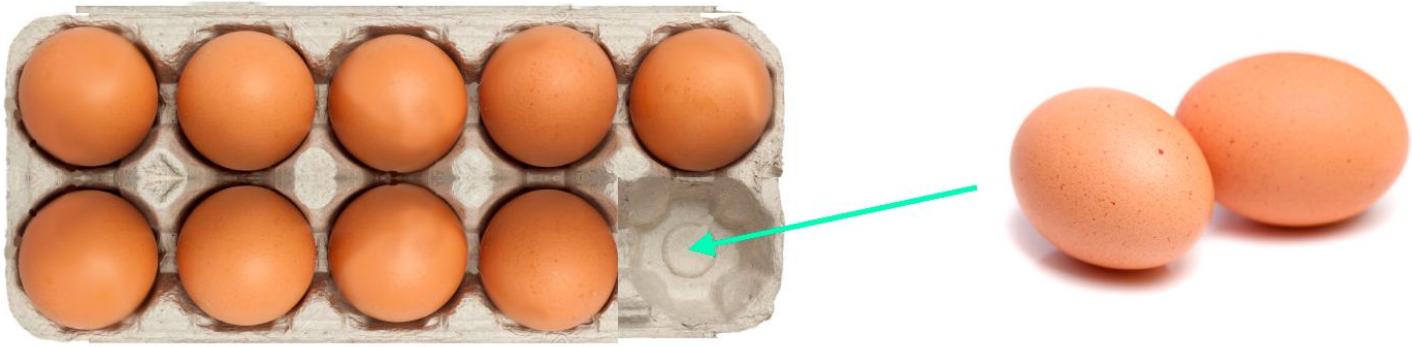
Emitter Size = 10

Emit Count = 2

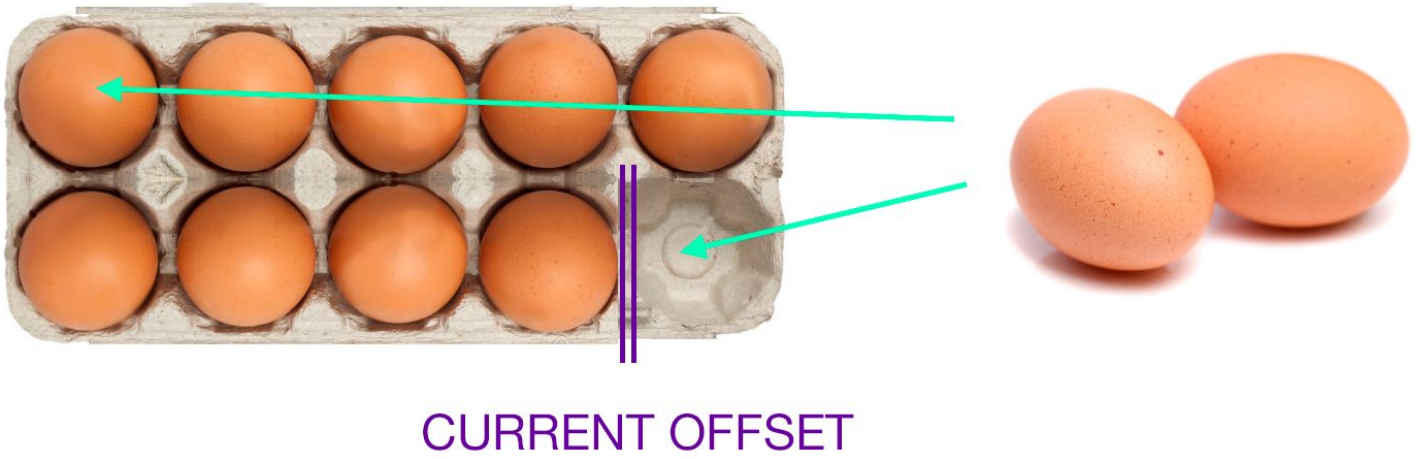
EMITTERS



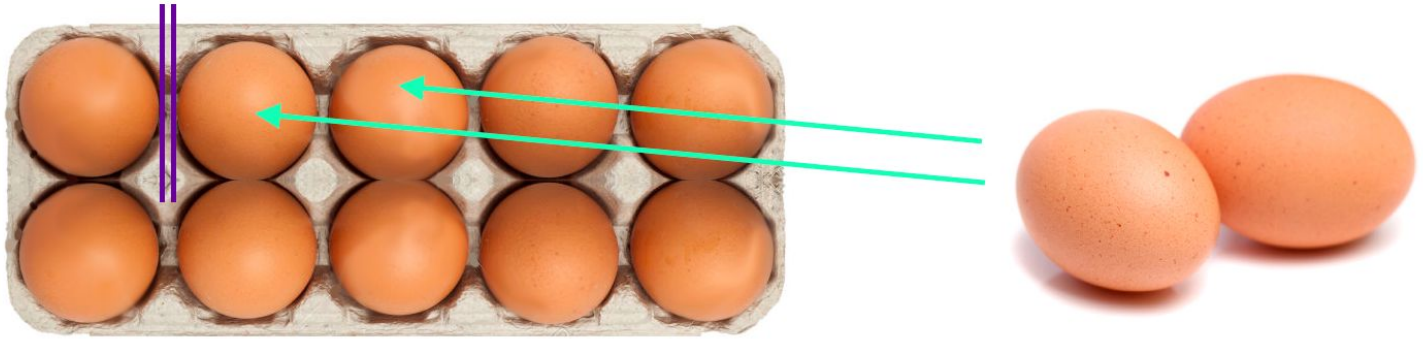
EMITTERS



EMITTERS (Force Mode = 1)

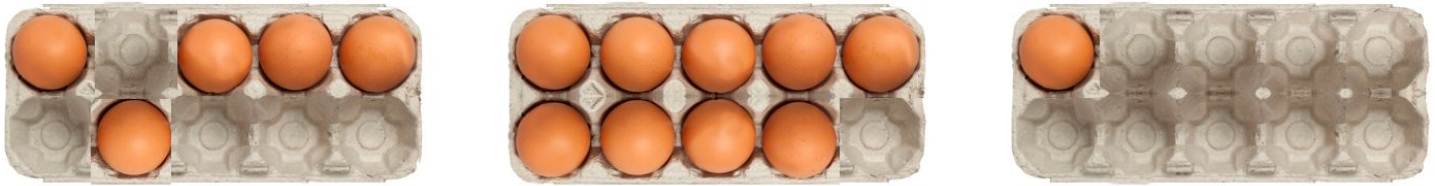


EMITTERS (Force Mode = 1)



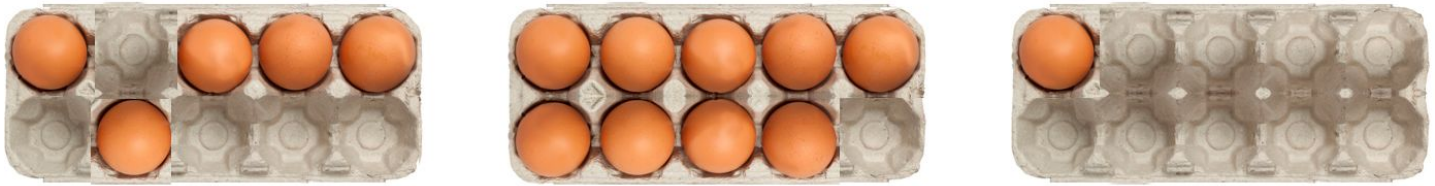
CURRENT OFFSET

PARTICLESYSTEM BUFFERS



Particle Count = 3 Emitter x 10 Particles = 30

PARTICLESYSTEM BUFFERS

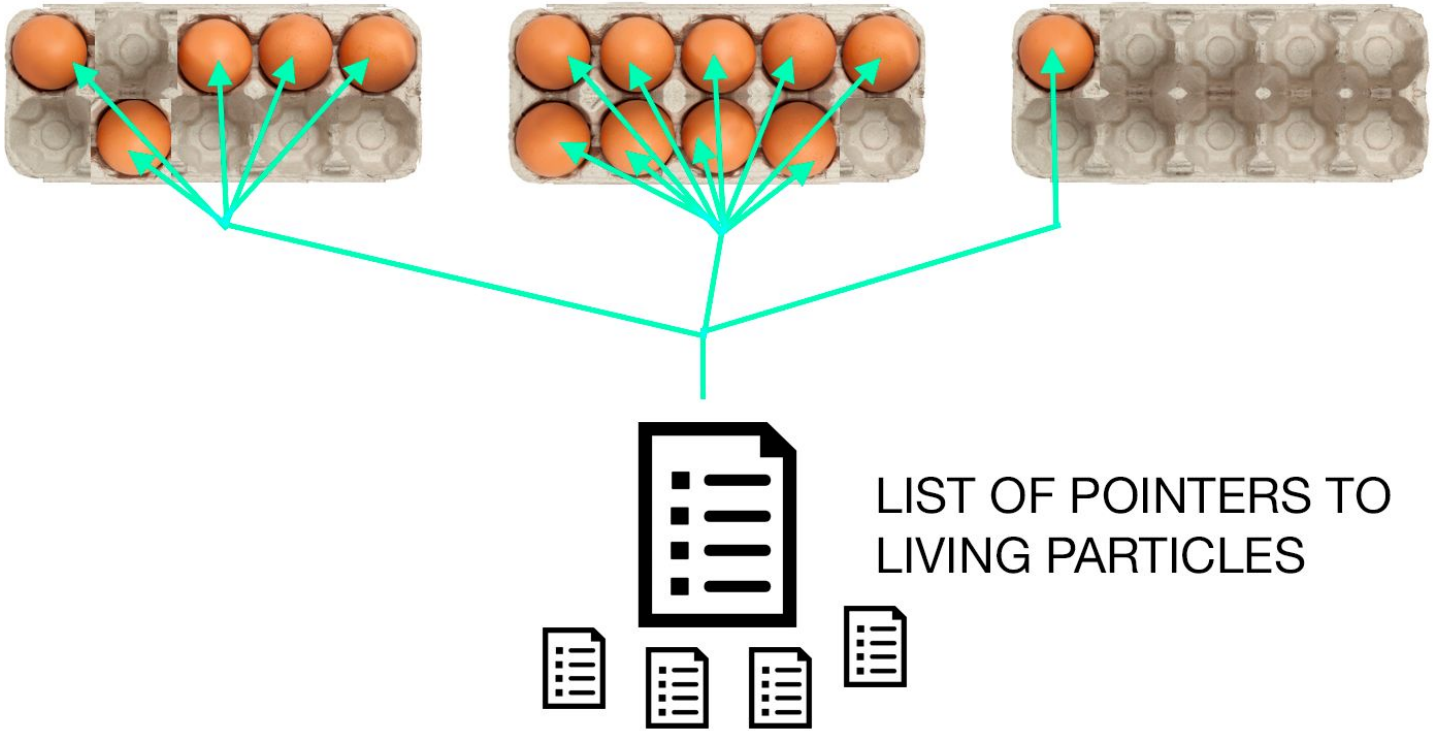


Particle Count = 3 Emitter x 10 Particles = 30

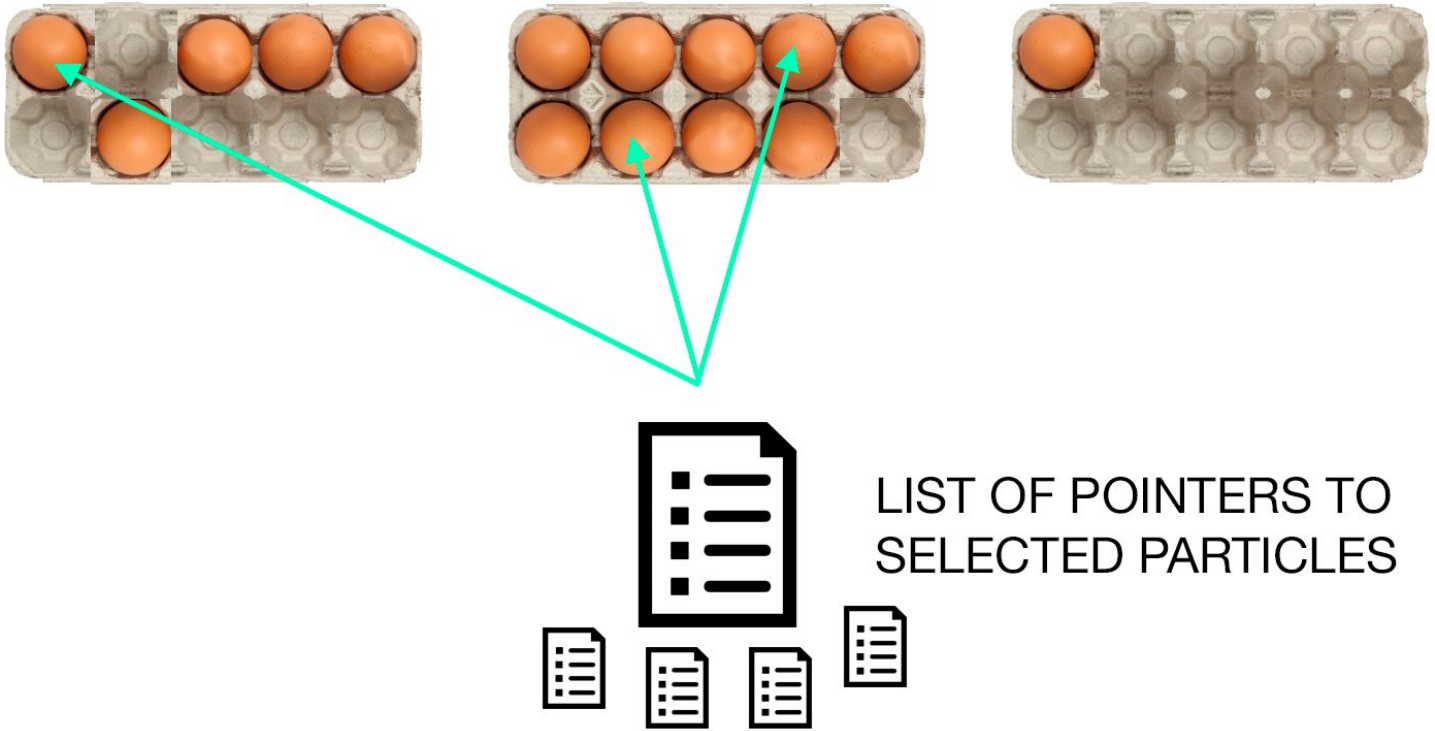
 = float3 position; float lifespan; float3 velocity; float age; float3 force;

Stride = 4 Bytes x 11 Values = 44 Byte
Memory Usage = 30 Particles x 44 Byte = 0.0013 MB

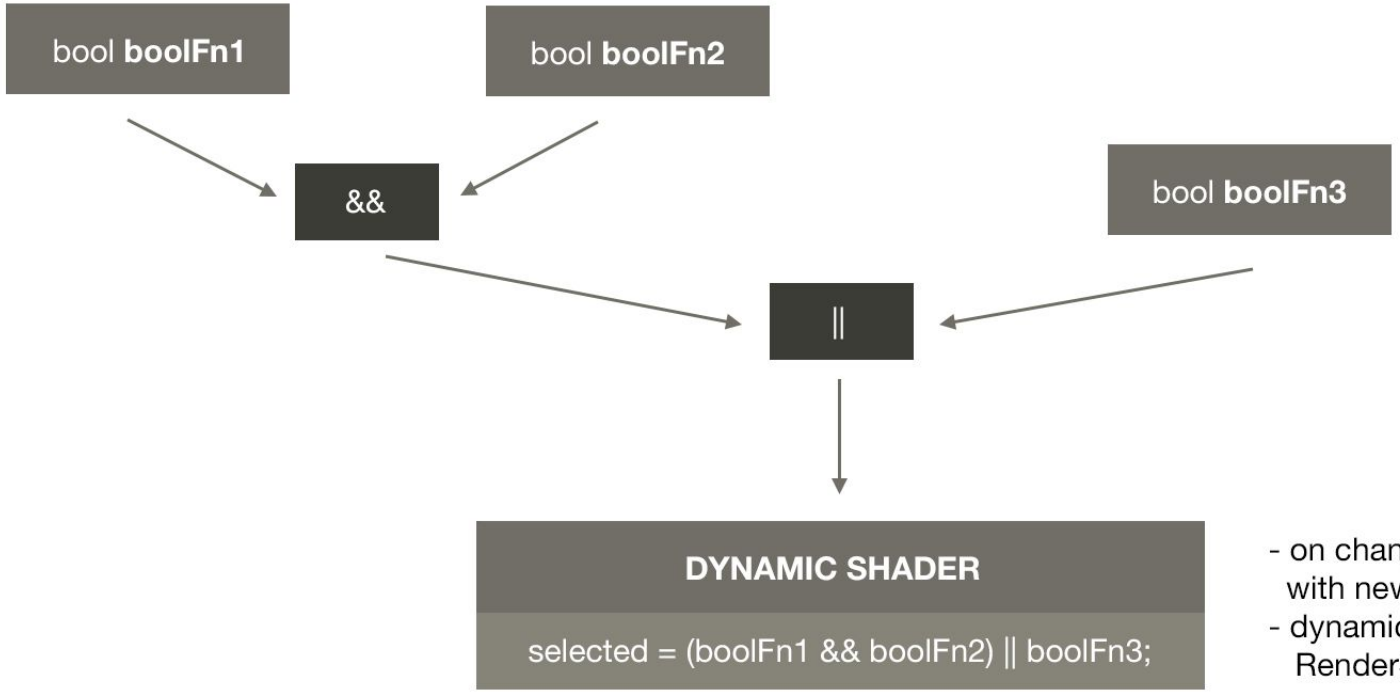
PARTICLESYSTEM BUFFERS



PARTICLESYSTEM BUFFERS



SELECTIONS



- on change shader recompiles with new logic expression
- dynamic values are passed via RenderSemantics/ResourceSemantics

COMMON MISTAKES / BEST PRACTISE / HINTS

- doublecheck if all particlesystem-enums of emitters/selectors/effects are equal to the name of the particle system (if you changed the name from default)
- sizes of emitters have to be static (otherwise the particlebuffer resets every frame)
- if nothing works as expected: use Sift (Buffer) + ReadBack nodes to debug the buffer data
- all attributes (also custom attributes) are initialized with 0 by emitters