Welcome
Topics

• Simulation
• Shading
• Creation
• Shadow Casting
• Summary
Introduction

Our Renderer

- Artist Node-Based Shader Editing
  - Our Rendering Engine running in Maya
  - Guerrilla Custom Shader Nodes
- Deferred Shading
- Targeting 30 fps at 1080p / 4K
Introduction

Horizon Zero Dawn

• Open World Action RPG
• Day and Night Cycle
• Post - Post Apocalyptic Northern America
• ‘BBC’ Nature
• Robotic Wildlife
Our first steps...

Killzone Shadow Fall:
- First Real Test in Production
- Quality and Performance
- Placement System Needed!
Placement System

2017 GDC Talk – Jaap van Muijden,
GPU-Based Procedural Placement in Horizon Zero Dawn

• Quick iterations
• Large variety and Believable look
• Art Directable
  • Data driven
  • Deterministic
  • Locally stable
Simulation

Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Simulation

Making Foliage Move

All Movement is driven by a Global Wind Force Field

- Compute Shader: around 150 microseconds
- Local Around the Player / Camera
- Baked-in Physical Properties
  - Supports up to 4 Different Categories of Spring Settings
Making Foliage Move

Categories of Spring Settings:
- Trees
  - Vegetation_Shader
- Plants
  - Vegetation_Shader
- Grasses
  - Vegetation_Grass_Shader
- Special
  - Sampled by Banners, Tarps, Canvas, etc.
Trees and Plants
Simulation

Vertex Program: Trees

Three Levels of Motion
Vertex Program: Trees

Three Levels of Motion:

• Tree Movement (Bend)
  • Height of the Object
Simulation

Vertex Program: Trees

Three Levels of Motion:

• Tree Movement (Bend)
  • Height of the Object

• Branch Movement
  • Distance to Trunk
Vertex Program: Trees

Three Levels of Motion:

• Tree Movement (Bend)
  • Height of the Object
• Branch Movement
  • Distance to Trunk
• Leaf Movement
  • Distance to Branch
Three Levels of Motion:

- Tree Movement (Bend)
  - Height of the Object
- Branch Movement
  - Distance to Trunk
- Leaf Movement
  - Distance to Branch
- Other Data
  - Index / Offset
Simulation

Vertex Program: Trees

Three Levels of Motion:

- Tree Movement (Bend)
  - Height of the Object
- Branch Movement
  - Distance to Trunk
- Leaf Movement
  - Distance to Branch
- Other Data
  - Index / Offset
  - Baked Ambient Occlusion
Simulation

Vertex Program: Trees

• Sample from our Global Wind at Object Center
  • Drives the Bend of the Tree
    • Rigidity
  • Drives the Bend of the Branches
    • Bend
    • Sway
    • Lift
• The length of the Sampled Force drives a tiny 3D Simplex Noise Texture (16x16x16) that we use for the motion of our Leaves.
  • Amount
Vertex Program: Plants

- (Most) Plants: Remove Bend
- Ramp – Soft Clamp Formula:
  - \( f(x) = \frac{x \cdot (\text{amount} + 1)}{x + \text{amount}} \)
Grasses

Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Vertex Program: Grass

Grass has three LODs:

- **LOD1**
  - 20-36 triangles
  - High shader

- **LOD2**
  - 10-18 triangles
  - High shader

- **LOD3**
  - 10-18 triangles
  - Low shader
Simulation

Vertex Programs: Making Grass Move

• Our Global Wind
Simulation

Vertex Programs: Making Grass Move

- Our Global Wind
- Ambient Motion
  - Large Scale Motion:
    \[
    X = (2 \times \sin(1 \times (ObjectCenterX + ObjectCenterY + ObjectCenterZ + Time))) + 1
    \]
    \[
    Y = (1 \times \sin(2 \times (ObjectCenterX + ObjectCenterY + ObjectCenterZ + Time))) + 0.5
    \]
    \[
    Z = 0
    \]
Vertex Programs: Making Grass Move

- Our Global Wind
- Ambient Motion
  - Large Scale Motion:
    
    \[
    X = (2 \times \sin (1 \times (ObjectCenterX + ObjectCenterY + ObjectCenterZ + Time))) + 1
    \]
    
    \[
    Y = (1 \times \sin (2 \times (ObjectCenterX + ObjectCenterY + ObjectCenterZ + Time))) + 0.5
    \]
    
    \[
    Z = 0
    \]
  - Small Scale Motion:
    
    \[
    Disp = (0.065 \times \sin (2.650 \times (PointWX + PointWY + PointWZ + Time)))
    \]
    
    \[
    \ast \ NormalObject \ast \ (1, 1, 0.35)
    \]
Simulation

Vertex Programs: Making Grass Do More

• On top of the motion, we also add:
  • Camera Based Tilting

\[ \text{Disp} = \text{[Vec3 (0, 1, 0) View to Object Space]} \cdot \text{ObjectPosZ} \]
Vertex Programs: Making Grass Do More

- On top of the motion, we also add:
  - Camera Based Tilting

\[
\text{Disp} = \text{[Vec3 (0, 1, 0) View to Object Space]} \ast \text{ObjectPosZ}
\]
Vertex Programs: Making Grass Do More

- On top of the motion, we also add:
  - Ground Hugging

\[ \text{Disp} = (0, 0, (\text{Sampled Height from Height map} - \text{ObjectCenterZ})) \]
Vertex Programs: Making Grass Do More

• To facilitate better Lodding, over distance:
  • we scale the whole animation part down
    \[\text{Displacement} = \text{[Percentage of All Animation]} \text{ based on Distance to Camera}\]
  • we vertically push the vertices of the mesh down
    \[\text{Displacement} = \text{[Percentage of Object Height]} \text{ based on Distance to Camera}\]
Shading – Alpha

Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Shading

Pixel Program: Alpha

• Depth Only Pass:
  • Very Cheap Depth Only Shader

• Geometry Pass:
  • Depth Compare (Depth is Equal)
  • Zero percent Overdraw!
45.560 %
Shading

Pixel Program: Alpha

Alpha Testing, Initial HZD setup:
• Alpha Textures are Signed Distance Textures
• Artists controlled size in Shader
• Artists controlled over Distance Quality
• Snow(flakes) in our Shader adjusted the Alpha
### Scalpel GPU Profiler

<table>
<thead>
<tr>
<th>Draw Hierarchy</th>
<th>Samplers</th>
<th>Shader state</th>
<th>Render state</th>
<th>Render targets</th>
<th>Name</th>
<th>Triangles</th>
<th>Tris/Physics</th>
<th>Cycles/Physics</th>
<th>Instances</th>
<th>DrawCalls</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Shadows</td>
<td>9.58%</td>
<td>1,693,254</td>
<td>0.00</td>
<td>0.00</td>
<td>4,554</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Geometry</td>
<td>2.14%</td>
<td>964,806</td>
<td>0.00</td>
<td>0.00</td>
<td>9,725</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DepthPrim</td>
<td>6.16%</td>
<td>968,006</td>
<td>0.00</td>
<td>0.00</td>
<td>9,725</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DepthOnly</td>
<td>6.16%</td>
<td>1,000</td>
<td>0.00</td>
<td>0.00</td>
<td>9,725</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pre材</td>
<td>0.00%</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Legend
- **Pan Area**: ALT + Mouse Drag
- **Zoom Area**: Mouse Drag
- **Zoom Bar**: Shift + Mouse Drag
- **Zoom All**: Ctrl + Mouse Drag
- **Packet to TTY/Packet**: Right Mouse Click
- **Record**: Ctrl + R
- **Send Frame**: Ctrl + Shift + R
- **Live View**: F11

### Selection
- **Same Name**
- **Same Name + Context**

### Table Controls
- **Row Select**
- **Page Skip**
- **First/Last Row**
- **Column Select**
- **Use Item**
- **Expand All**
- **Collapse All**
- **Print Table**
- **Save to File**

### Key Shortcuts
- **Up/Down**
- **Page Up/Down**
- **Ctrl + Tab**
- **Edit**
- **Ctrl + Enter**
- **ALT + Enter**
- **T**
Pixel Program: Alpha

Alpha Testing, Solution:

• No more adjusting the Alpha in the Shader!

• ‘Custom Coverage’ Algorithm to build a better Mip Chain:
  • Calculate coverage of the input image (after alpha testing).
  • Generate a regular Mip-chain.
  • For each Mip:
    • Generate a histogram of this Mip after bilinear up sampling. In our case to 4096x4096.
    • Find the point P in the histogram that corresponds to the original coverage.
    • Scale the pixels by 0.5 / P so that P ends up at our alpha testing value of 0.5.
Pixel Program: Alpha

Alpha Testing Optimization:
- Make the Alpha fit in Cache!
- In our case, Small Assets and Grass

256x128 pixel Alpha Texture
- Need really good Anti-Aliasing!
Anti-Aliasing

SIGGRAPH 2017

Giliam de Carpentier - Guerrilla Games
Kohei Ishiyama - Kojima Productions

Decima Engine: Advances in Lighting and AA

• Typically 4 Samples in total
• Always two Samples per rendered pixel per frame
• At most 1 millisecond per frame at 1080 / PS4
Shading

Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Shading

Pixel Program: G-Buffers

Our Vegetation Shaders write to Geometry Buffers:

- Normal
- Albedo
- Roughness
- Reflectance
- Translucency Amount
- Translucency Diffusion
- Depth
- Motion Vectors - *(if placed by the Placement System)*
Shading

Pixel Program: Vegetation Textures

Vegetation Textures:
- Alpha
- Tangent Space Normal
- Albedo
- Translucency Amount
- Mask
- Ambient Occlusion
  - Not on Grass!
Shading

Pixel Program: Vegetation Textures

Vegetation Textures:
- Alpha
- Tangent Space Normal
- Albedo
- Translucency Amount
- Mask
- Ambient Occlusion
  - Not on Grass!

Get packed into:
- BC7: NMT / MSK / AO
- BC7: CLR / TRA
- BC4: ALPHA
Shading

Pixel Program: Normals

Our Normals, when Rendered Double Sided are:

- Flipped Correctly in Tangent Space
  - Most Plants, lots of Small Trees, most Pine Trees
Our Normals, when Rendered Double Sided are:

- Flipped Correctly in Tangent Space
  - Most Plants, lots of Small Trees, most Pine Trees
Shading

Pixel Program: Normals

Our Normals, when Rendered Double Sided are:

- Flipped Correctly in Tangent Space
  - Most Plants, lots of Small Trees, most Pine Trees
- Flipped ‘Incorrectly’
  - Adjusted Vertex Normals
  - \( \text{Abs()} \) the \( Z \) components of our Viewspace Normal
  - Grass and most of our Trees Canopies
Shading

Pixel Program: Normals

Our Normals, when Rendered Double Sided are:

• Flipped Correctly in Tangent Space
  • Most Plants, lots of Small Trees, most Pine Trees

• Flipped ‘Incorrectly’
  • Adjusted Vertex Normals
  • $\text{Abs}()$ the Z components of our Viewspace Normal
  • Grass and most of our Trees Canopies
Our Normals, when Rendered Double Sided are:

- Flipped Correctly in Tangent Space
  - Most Plants, lots of Small Trees, most Pine Trees
- Flipped ‘Incorrectly’
  - Adjusted Vertex Normals
  - Abs() the Z components of our Viewspace Normal
  - Grass and most of our Trees Canopies
Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn

'NORMAL' VERTEX NORMALS

CUSTOM VERTEX NORMALS
Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Pixel Program: Normals
Pixel Program: Normals

Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Pixel Program: Albedo

Colorization Texture Array:
- Based on Asset Type
  - Artist driven
    - U-Component
  - Based on World Data
    - Erosion, Flow, Closeness to Water baked into 512x512 Worlddata Texture
    - V-Components
- Based on Ecotope
  - Place in the World
    - W-Component
In our shader, we colorize most of our Vegetation:

- Two Textures drive colorization
  - Albedo
  - Mask
- Illuminate Blend Function
  - Result = Texture * ( 2 * Colorize * Mask + 1 – Mask)
Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Shading

Pixel Program: Roughness / Reflectance

- Reflectance – Fixed at 4% Dielectric
  - Baked Ambient Occlusion and Occlusion Texture
- Roughness – Artist controlled
  - Baked Ambient Occlusion, Occlusion Texture and Translucency Texture
- Grass – Similar Value Setup, but:
  - Influenced by Translucency Texture
Shading

Pixel Program: Translucency

Our Translucency is the product of the following:

- Amount of light hitting the object from behind
- Angle between the camera, the lit pixel on screen and the light source
- ‘Max Luminance’ of Albedo Color

- Surface Thickness
- Pre-computed Ambient Occlusion
- Boost for Artistic Purposes
Asset Creation

Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Asset Creation

Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Creation Process

- Build ‘Speed Model’
  - Maya / SpeedTree / Photoshop
Creation

Process

• High Detail Creation
  • Maya / SpeedTree / Photoshop
Creation

Process

• Baking into UV Space
  • Maya / SpeedTree / Photoshop
Creation

Process

- Build LOD Chain of Components
  - Maya
Creation Process
Creation

Process

• Build Asset out of Components
  • SpeedTree
Creation

Process

• Build Asset out of Components
  • SpeedTree
Creation
Process

• SpeedTree Export to Maya
  • Houdini

• Asset Setup / Export to Game
  • Maya
Creation

Assets: Trees

Trees have five LODs:

- **LOD1**
  - ~10000 triangles
  - High shader
- **LOD2**
  - ~2600 triangles
  - High shader
- **LOD3**
  - ~1200 triangles
  - Low shader
- **LOD4**
  - Low shader + fading in billboard
  - ~200 + 12 triangles
- **LOD5**
  - Billboard shader
  - 12 triangles
Creation

Assets: Plants

- Plants have three LODs:
  - LOD1
    - ~2000 triangles
    - High shader
  - LOD2
    - ~800 triangles
    - High shader
  - LOD3
    - ~140 triangles
    - Low shader
  - LOD4
    - 8 triangles
    - Billboard shader
Shadow Casting

Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Shadow Casting

For sun shadows, we use four cascades:

- **Compartment Cascade**
  - 1024 x 1024 map - covers Aloy only

- **Cascade 0**
  - 2048 x 2048 map - to 8m away from camera

- **Cascade 1**
  - 2048 x 2048 map - to 80m away from camera

- **Distant cascade – height-field-based system**
  - 1536 x 1536 map - 1.5km x 1.5km region around the camera.
Shadow Casting

For sun shadows, we use four cascades:

- **Compartment Cascade**
  - 1024 x 1024 map - covers Aloy only
- **Cascade 0**
  - 2048 x 2048 map - to 8m away from camera
- **Cascade 1**
  - 2048 x 2048 map - to 80m away from camera
- **Distant cascade** – height-field-based system
  - 1536 x 1536 map - 1.5km x 1.5km region around the camera.

*Smooth Alpha Blended Crossfade between Cascades!*
Shadow Casting Vegetation

- Separate Visual Meshes and Shadow Casting Meshes
- Specific Meshes, Textures and Shaders for Cascade 0 and 1
  - S1:
    - LOD3 Visual Mesh
    - Alpha Tested, Animated
    - Depth Only Shader
  - S2:
    - Even less triangles
    - Non-Alpha Tested, Non-Animated
    - Depth Only Shader
Summary

Guerrilla Games – GDC 2018 – Between Tech and Art: The Vegetation of Horizon Zero Dawn
Summary

What worked for us?
- Depth Prime
- Custom Mip Chain
- LOD up, not down
- Shadow Casters separate
- Placement System
- All In-house
Thank You!
References

Giliam de Carpentier (Guerrilla Games) & Kohei Ishiyama (Kojima Productions)
SIGGRAPH 2017 - Decima Engine: Advances in Lighting and AA

Jaap van Muijden
2017 GDC Talk - GPU-Based Procedural Placement in Horizon Zero Dawn

Renaldas Zioma
GPU Gems 3 - Chapter 6: GPU-Generated Procedural Wind Animations for Trees