DirectX 11 Rendering & NVIDIA GameWorks in Batman: Arkham Origins
Colin Barré-Brisebois (Lead Rendering Programmer)
AGENDA

PREFACE
   ➔ Motivations

PART 1-DEFORMABLE SNOW
   Excerpts from GDC 2014
   ➔ Novel technique for surfaces covered with fallen deformable snow
   ➔ Enhanced for PC using DX11

PART 2-CROMA SUBSAMPLING
   ➔ Various pipeline enhancements

PART 3-NVIDIA GAMEWORKS
   ➔ Enhanced game features using NVIDIA GameWorks
PART 1
DEFORMABLE SNOW RENDERING
Deformable Snow Surfaces

ENHANCE THE WORLD WITH DYNAMICS OF DEFORMABLE SNOW

THREE REQUIREMENTS:
1. Iconic visuals of deformable snow
2. Organic deformation from walking, falling, sliding, fighting and more
3. Low memory usage and low performance cost for an open world game

[Barré-Brisebois 2014]
Iconic & Organic Deformable Snow

Previous Work?

[ST-AMOUR 2013] (ASSASSIN’S CREED 3)
[EDWARDS 2012] (JOURNEY)

- RAYCAST ON A TERRAIN / MODIFY TERRAIN MESH.
  - We don’t have terrain. We have rooftops and streets.
  - Besides, we don’t want to add raycasts.

- REQUIRES VARIABLE TRIANGLE DENSITY FOR VISUALLY CONVINCING VERTEX DISPLACEMENT IN ALL CASES
  - PC DX11 with tessellation is great... but what about consoles?
Our Approach (1/)

- **GENERATE DISPLACEMENT HEIGHTMAPS AT RUNTIME**
  - Snow deformation is a semi-low frequency detail effect
  - Cheap approximation works with footsteps & more
  - Great performance, and low memory usage

- **CONSOLES: VIRTUAL DISPLACEMENT VIA RELIEF MAPPING**
  - Minimal taps. No “swimming”
  - Independent of triangle density

- **PC: DIRECTX 11 VERSION WITH TESSELLATION**
Our Approach (2/)

- **GOTHAM HAS MANY ROOFTOPS AND STREETS**
  - Dynamically alloc/dealloc heightmaps based on size, player/AIs and visibility

[Barré-Brisebois 2014]
Generating the Heightmap?

- **RENDER SNOW-AFFECTING OBJECTS LOOKING FROM UNDER THE SURFACE USING AN ANKLE-HIGH ORTHOGONAL FRUSTUM**
  1. Clear to black
  2. Render actors in white
  3. Filter and accumulate (ping/pong) in a texture

- **ANYTHING IN THAT ZONE WILL AFFECT THE HEIGHTMAP**
  1. Feet, hands
  2. Sliding, throwing a thug to the ground...
Ankle-high Orthogonal Frustum
Video!
Update Loop

FOR EVERY ACTIVE* SNOW SURFACE

1. Figure out if surface-affecting object is on the surface
   - We use a quad tree look-up rather than keeping an actor list for each surface
2. Override materials on all parts
   - Simple white material
3. Render actors
4. Process/Accumulate with custom post-process chain

[Barré-Brisebois 2014]
Heightmap Accumulation & Render

- **STAGE 1 - GET RESULTS & SMALL BLUR**
  - 4-tap bilinear Poisson

- **STAGE 2 - ADD TO EXISTING HEIGHTMAP**
  - During this stage, you can also subtract a small value to the heightmap to make snow gradually replenish (since it’s snowing) 😊

- **STAGE 3 - SHADING**
Stage 3 - Shading (1/)

SNOW SURFACES HAVE 2 MATERIAL OPTIONS:

1. Basic Snow Material
   - Active when surface is not being deformed
   - Shows new / clean / untouched snow, cheaper

2. Deformable Snow Material
   - Two stages: non-deformed or fully flattened snow
   - Non-deformed part the same as Basic Snow Material
   - Fully flattened shows rooftop tiles / concrete.
   - Blends both stages using heightmap & Relief Mapping

[Barré-Brisebois 2014]
Stage 3 - Shading (2/)

Non-deformed Snow

Flattened Snow

[Barré-Brisebois 2014]
BLENDING MATERIAL STAGES

- For diffuse & spec, simple lerp
  - Also, tint diffuse with sky color in transition area to fake SSS
- For normals, blend using Reoriented Normal Mapping
  [Barré-Brisebois & Hill 2012]
  - Normals are not colors.
  - You can’t lerp/overlay between directions!
- Used in game to:
  - Blend the snow detail normal and the macro “wave” snow normal
  - Add detail normal maps everywhere
float3 t = tex2D(BaseNormal, uv) * float3(2, 2, 2) + float3(-1, -1, 0);
float3 u = tex2D(DetailNormal, uv) * float3(-2, -2, 2) + float3(1, 1, -1);
float3 r = t * dot(t, u) / t.z - u;

[Barre-Brisebois & Hill 2012]
LINEAR INTERPOLATION
OVERLAY BLEND
REORIENTED NORMAL MAPPING

ADOBE, PLEASE ADD TO PHOTOSHOP!

[Barré-Brisebois & Hill 2012]
Add. Implementation Details

- **SURFACE UV ALIGN WITH ORTHO FRUSTUM**
  - 0-1 range, simplifies heightmap-to-displacement

- **SCALED WORLD-SPACE HEIGHTMAP RES.**
  - \( \text{Min}(512, \frac{1}{4} \times (\text{SurfaceX}, \text{SurfaceY})) \)
  - Tries to keep texels “square”
  - Doesn’t need to be high-res, looks better in lower resolutions
  - Must scale Relief Mapping parameters

- **SPLIT RENDER & TICK OF ACTIVE SURFACES**
  - Snow surface where Batman stands has priority
  - We only render 2 surfaces/frame (tweakable, distance-based)

[Barré-Brisebois 2014]
DIRECTX 11 WITH TESSELLATION (1/)

- DEVELOPED WITH OUR FRIENDS@NVIDIA (EVGENY MAKAROV)
- ACCURATE DISPLACEMENT BASED ON DEPTH
  - Capture the height field like a z-buffer
  - Two channels:
    - Minimum height field
    - Projected displacement
  - Allows for additive capture, smoother results and deformable snow banks!

- TESSELLATED VERSION ADDS DETAILED DISPLACEMENT CALCULATED FROM THE NORMAL MAP
  - Macro deformation breaks surface uniformity
DirectX 11 With Tessellation (2/)

- Orthogonal Capture Frustum
- Minimum Height field
- Projected Displacement
- Final Surface (displaced)

Rooftop

[Barré-Brisbois 2014]
Without Tessellation (No Macro Deformation)
With Tessellation (Macro Deformation)
DirectX 11 With Tessellation (4/)

- RUNTIME DICING OF SNOW MESHES
- REAL GEOMETRY MEANS:
  - *Works with Dynamic Shadows*
    - Character shadows now follows the surface and shift with the deformation
    - Self shadowing & self-shading
  - *Works with dynamic ambient occlusion*
    - AO fills-in the trails
Performance & Memory

- **PERFORMANCE**
  - Heightmaps update < 1.0ms GPU on PS3/360

- **MEMORY**
  - 2 MB (360 / PS3 / WiiU)
    - Since we’re using low resolution heightmaps
    - This is flexible, but sufficient for our needs since we allocate/deallocate as the player flies in the world
  - 2-4 MB (FP16 vs FP32 on PC)

[Barré-Brisebois 2014]
Caveats / Issues?

- **RELIEF-MAPPED APPROACH**
  - Deformation looks great, but will never be as thick as tessellation. Replace with Parallax Occlusion Mapping?
  - Derive parametric AO from the heightmap?

- **TESSELLATED APPROACH**
  - When artists were working on content creation, displacement wasn't taken into account (pre-pass actors, open edges being visible, etc...)
  - Some meshes couldn't use tessellation as there were parts of geometry right under the snow, not supposed to be visible
Future Endeavors...

- Save the heightmaps and reload them?
- Use this technique for other cases, such as sand, mud, etc...
- Improvements to slopes with the depth-based approach
PART 2
CHROMA SUBSAMPLING IMPROVEMENTS
ARKHAM ORIGINS’ WORLD IS DOUBLE THE SIZE OF ARKHAM CITY
- More data, but same amount of storage...

USE CHROMA SUBSAMPLING
Chroma subsampling is the practice of encoding images by implementing less resolution for chroma information than for luma information, taking advantage of the human visual system's lower acuity for color differences than for luminance. ¹

THIS IS NOT NEW...
- Broadcasting
- Oh, and in games too! [White and Penty 2009]

¹ http://en.wikipedia.org/wiki/Chroma_subsampling
Chroma Subsampling? (2/)

$Y_{(W, H)}$ 

$CoCg_{(W/N, H/N)}$

$N \rightarrow$ Your art director’s tolerance to compression (2, 4...) 😊
Chroma Subsampling? (3/)

- USED EXTENSIVELY ON LIGHTMAPS
  - Unlike [White and Penty 2009], we used YCoCg
  - Pack two luma textures together instead of 3
    - Blue channel pollution from DXT1 compression is too significant in our case when packing 3
    - Improve quality by setting B channel to 0 and/or set the DXT weights
      - (R=0.5, G=0.5, B=0)

- USED EXTENSIVELY ON LODS
  - We have 2 (NEAR & FAR) levels of LODs, let’s compress more!

- OVERALL BETTER COMPRESSION RESULTS OVER VANILLA DXT1 LIGHTMAPS & LODS
Chroma Subsampling? (4/)

- CHROMA SUBSAMPLING CAN BE USED EXTENSIVELY...
  - We started implementing chroma frame buffer compression
  - Similar to [Mavridis & Papaioannou 2012]’s *The Compact YCoCg Frame Buffer*
  - Didn’t make it into the final game 😞
  - Nonetheless...

- WE SHOULD ALWAYS FIND MORE OPPORTUNITIES TO COMPRESS MORE, ESPECIALLY WHEN IT MAKES “VISUALLY SENSE”!
  - Next time! 😊
PART 3
NVIDIA GAMEWORKS ENHANCEMENTS
BATMAN: ARKHAM ORIGINS
FEATURED NVIDIA TECHNOLOGY

CONTACT-HARDENING SHADOWS
ENVIRONMENTAL CLOTH
PHYSX PARTICLES

HBAO+
SNOWPRINTS: TESSELATION

DEPTH OF FIELD
BLIZZARD CONDITIONS: TURBULENCE
TEARING CLOTH

BATMAN: TXAA, HBAO+, SOFT SHADOWS, CAPE TESSELLATION
TURBULENCE
VIDEO
BATMAN: ARKHAM ORIGINS
FEATURED NVIDIA TECHNOLOGY

- CONTACT-HARDENING SHADOWS
- ENVIRONMENTAL CLOTH
- PHYSX PARTICLES
- HBAO+
- TXAA, HBAO+, SOFT SHADOWS, CAPE TESSELLATION
- DEPTH OF FIELD
- BLIZZARD CONDITIONS: TURBULENCE
- SNOWPRINTS: TESSELLATION
- TEARING CLOTH
- TURBULENCE
DEPTH OF FIELD
PARTICLES
PARTICLE SHADOW MAPPING

Self-shadowed particles

PSM Enabled

Cast shadow

PSM Disabled
CAPE TESSELLATION (1/)
CAPE TESSELLATION (2/)
Summary

- A FAST AND LOW-MEMORY FOOTPRINT TECHNIQUE TO RENDER DEFORMABLE SNOW SURFACES
  - Adds a really nice level of interaction between players and the world
  - Depicts iconic & organic visuals of deformable snow
  - A good tessellation case for your DX11 game using minimal editing and art tweaks

- CHROMA SUBSAMPLING FOR YOUR ART PIPELINE IS DEFINITELY A VENUE TO CONSIDER

- HOW WE USED NVIDIA’S GAMEWORKS TO ENHANCE THE VISUALS OF SNOW BEHIND BATMAN: ARKHAM ORIGINS
Thank You!

Érick Bilodeau
David Massicotte
Sébastien Turcotte
Jimmy Béliveau
Olivier Pomarez
Philippe Bernard
Ryan Lewis
Marc Bouchard
Jean-Noé Morisette
Pierric Gimmig
Patrick Dubuc

Reid Schneider
Maggy Larouche
Miguel Sainz
Evgeny Makarov
Jon Jansen
Louis Bavoil
Holger Gruen
Andrei Tatarinov
Andrew Burnes
Jon Greenberg

NVIDIA

Questions?

colin.barrebrisebois@wbgames.com / @ZigguratVertigo
We’re Hiring! http://www.wbgamesmontreal.com
References (1/)

[Barré-Brisebois 2014]

[Barré-Brisebois & Hill 2012]

[Bavoil & Jansen 2013]

[Edwards 2013]

[Jansen & Bavoil 2011]
References (2/)

[Policarpo & Oliveira 2006]

[Mavridis & Papaioannou 2012]

[St-Amour 2013]

[White & Penty 2009]