Université Lumière Lyon 2

Institut de la communication

gamagora

Cours

Génération procédurale par l'exemple

Master 2 Programmation et Développement

C Le Bihan Gautier

C. Le Bihan Gautier - Université Lumière Lyon 2- Laboratoire LIRIS
corentin,gautier@universite-lyon,fr

Génération procédurale

Plan :

- Application de la génération procédurale
- Génération de routes
- L-system :
  - Génération d'arbres
  - Génération d'une ville
- Génération de terrains

https://www.youtube.com/watch?v=-d2-PtK4F6Y

- Une introduction par Kate Compton

INDEPENDENT GAMES SUMMIT

Practical Procedural Generation for everyone!

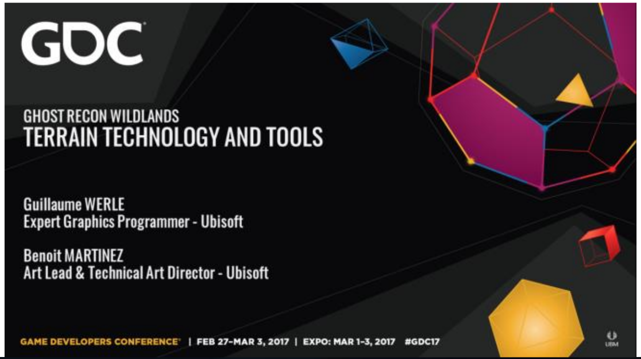
Kate Compton

independent mad scientist/inventor

GDC
GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17
GDC 17

https://www.youtube.com/watch?v=WumyfLEa6bU

Exemple Ghost Recon Wildlands



GDC

GHOST RECON WILDLANDS
TERRAIN TECHNOLOGY AND TOOLS

Guillaume WERLE
Expert Graphics Programmer - Ubisoft

Benoit MARTINEZ
Art Lead & Technical Art Director - Ubisoft

GAME DEVELOPERS CONFERENCE | FEB 27–MAR 3, 2017 | EXPO: MAR 1–3, 2017 #GDC17

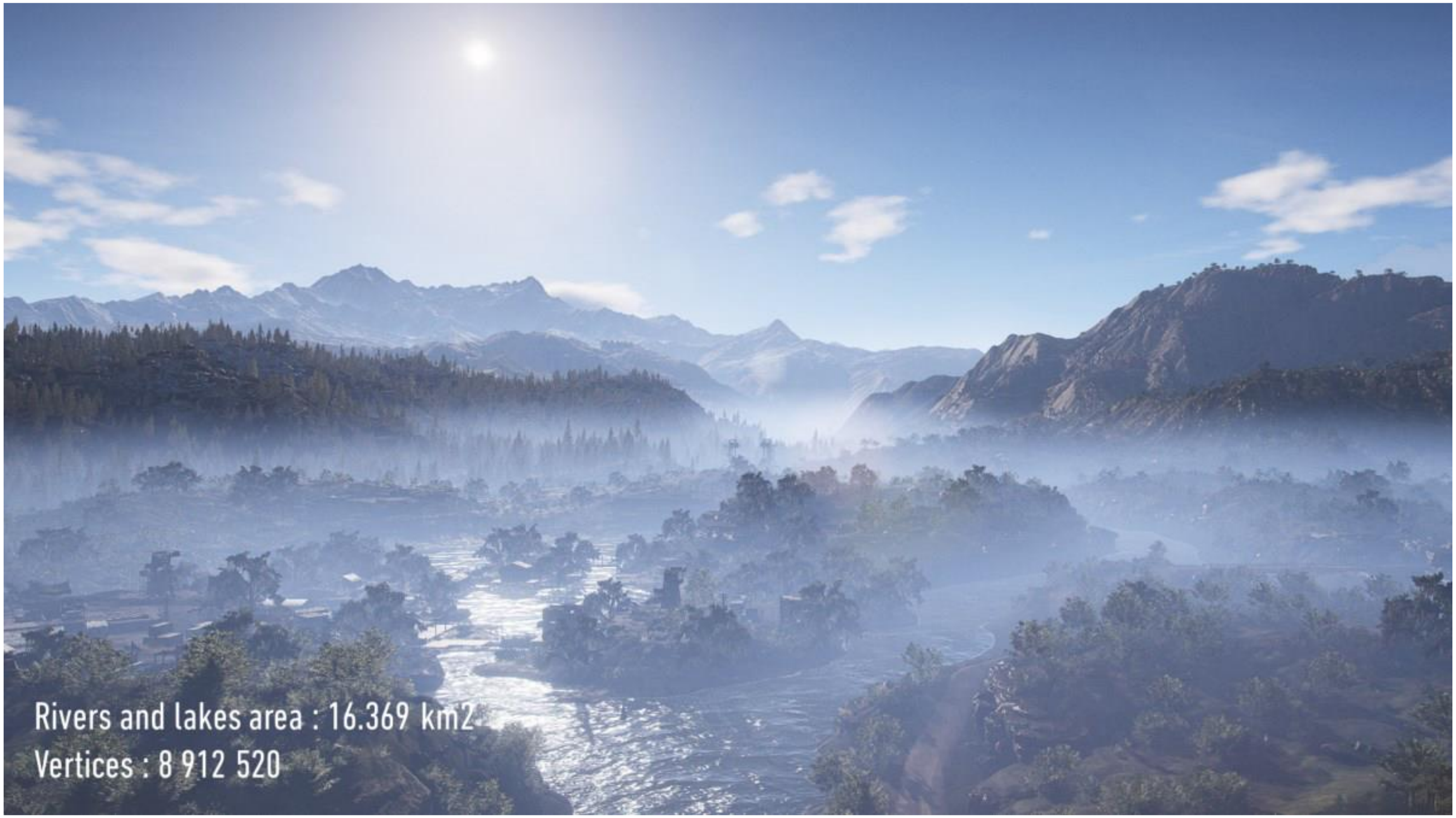
https://www.gdcvault.com/play/1024029/-Ghost-Recon-Wildlands-Terrain

- GhostReconWildlands is the largest action adventure open world ever made at Ubisoft

11 Biomes
140 materials

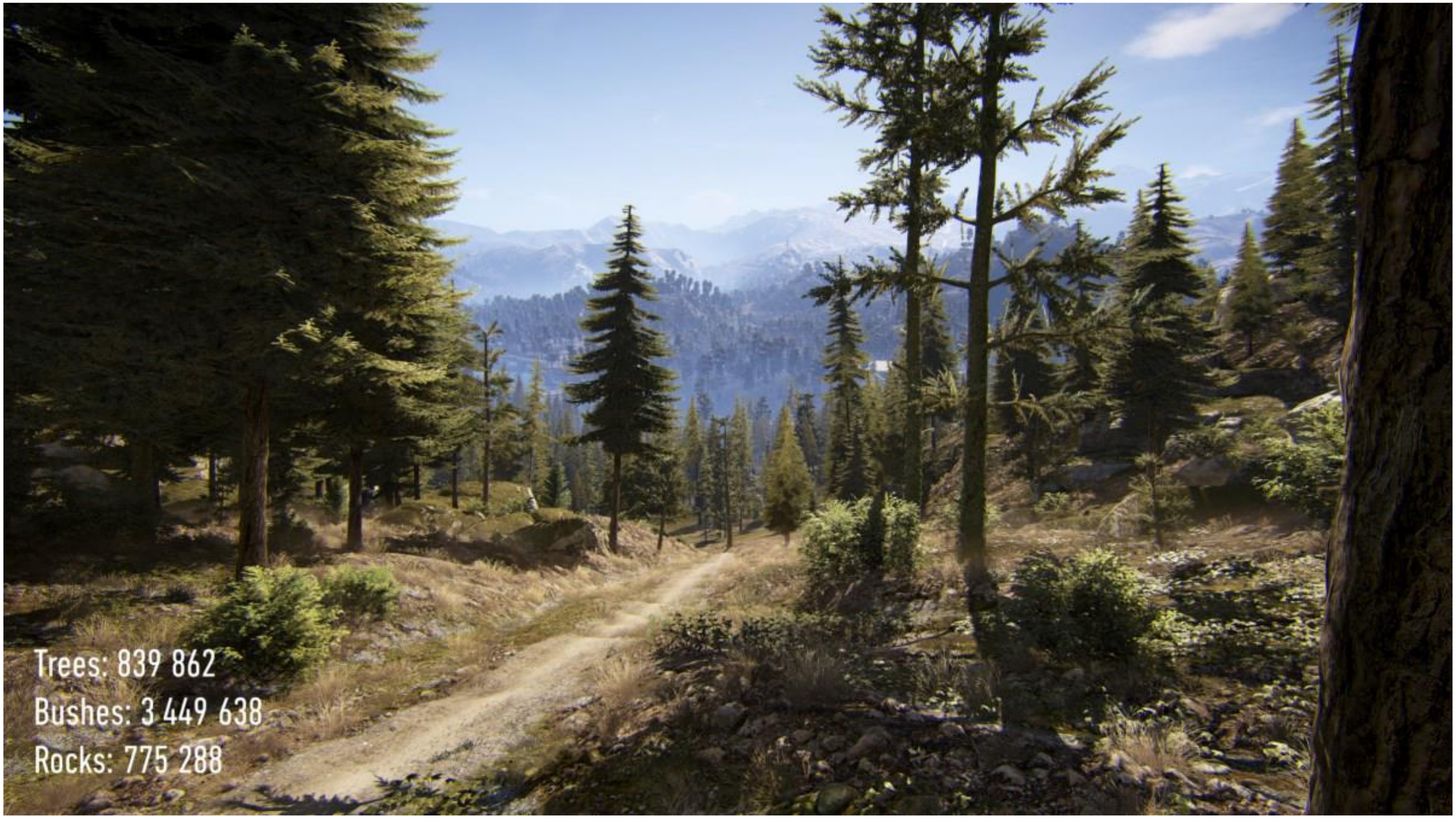
The game is set in Bolivia
We have a lot of diversity in the game, 11 completely different biomes

- Rivers and lakes area



Lakes, rivers and streams over a surface of 16km2

- Végétation et rochers



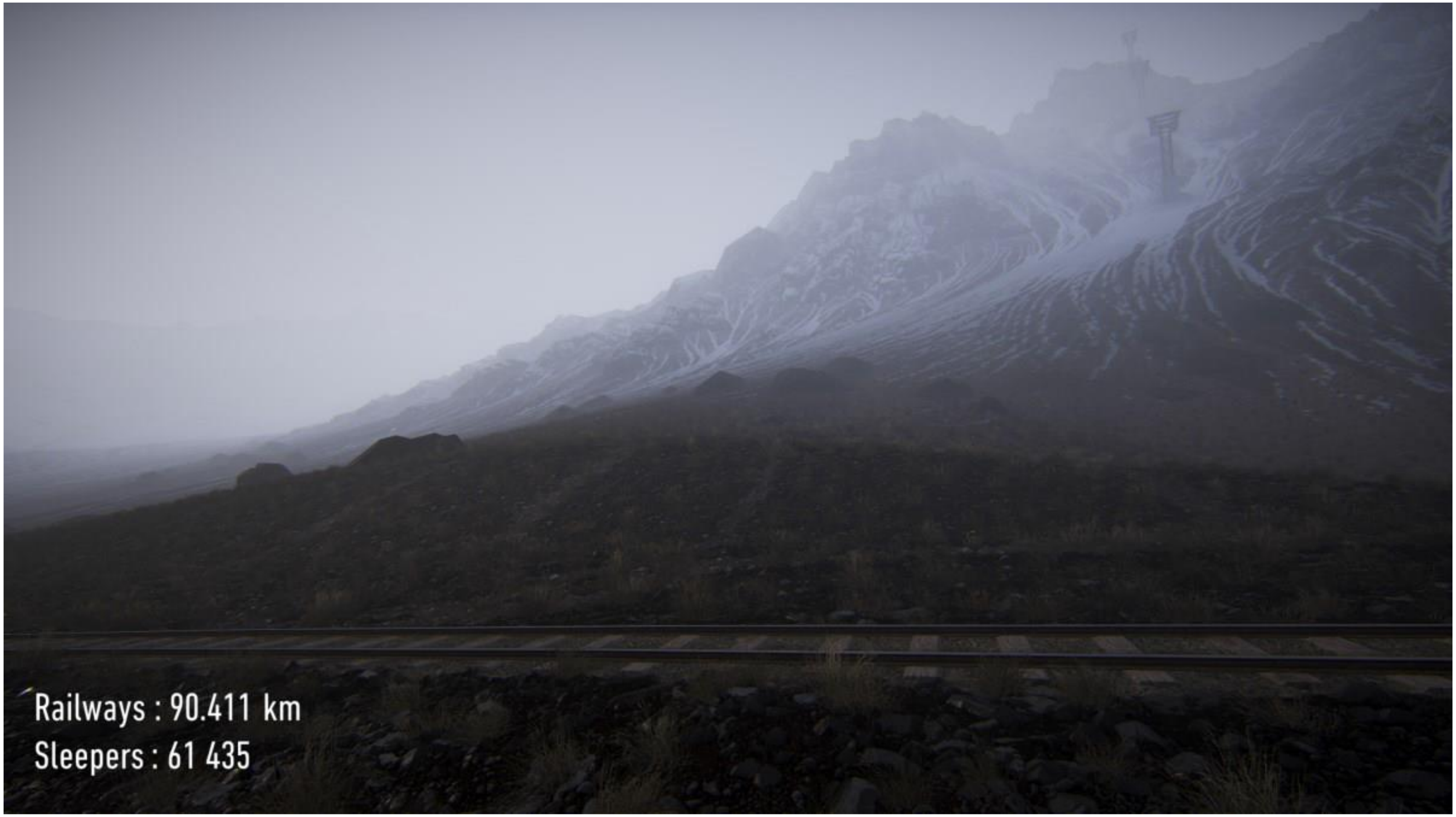
It’s mostly a natural environment with many forests
Millions of trees, bushes and rocks

- Génération de routes

Roads: 656.421 km
Footpaths: 2150.24 km
Crash Barriers: 10 712 modules
Traffic Signs: 781
Decals: 4 874 935

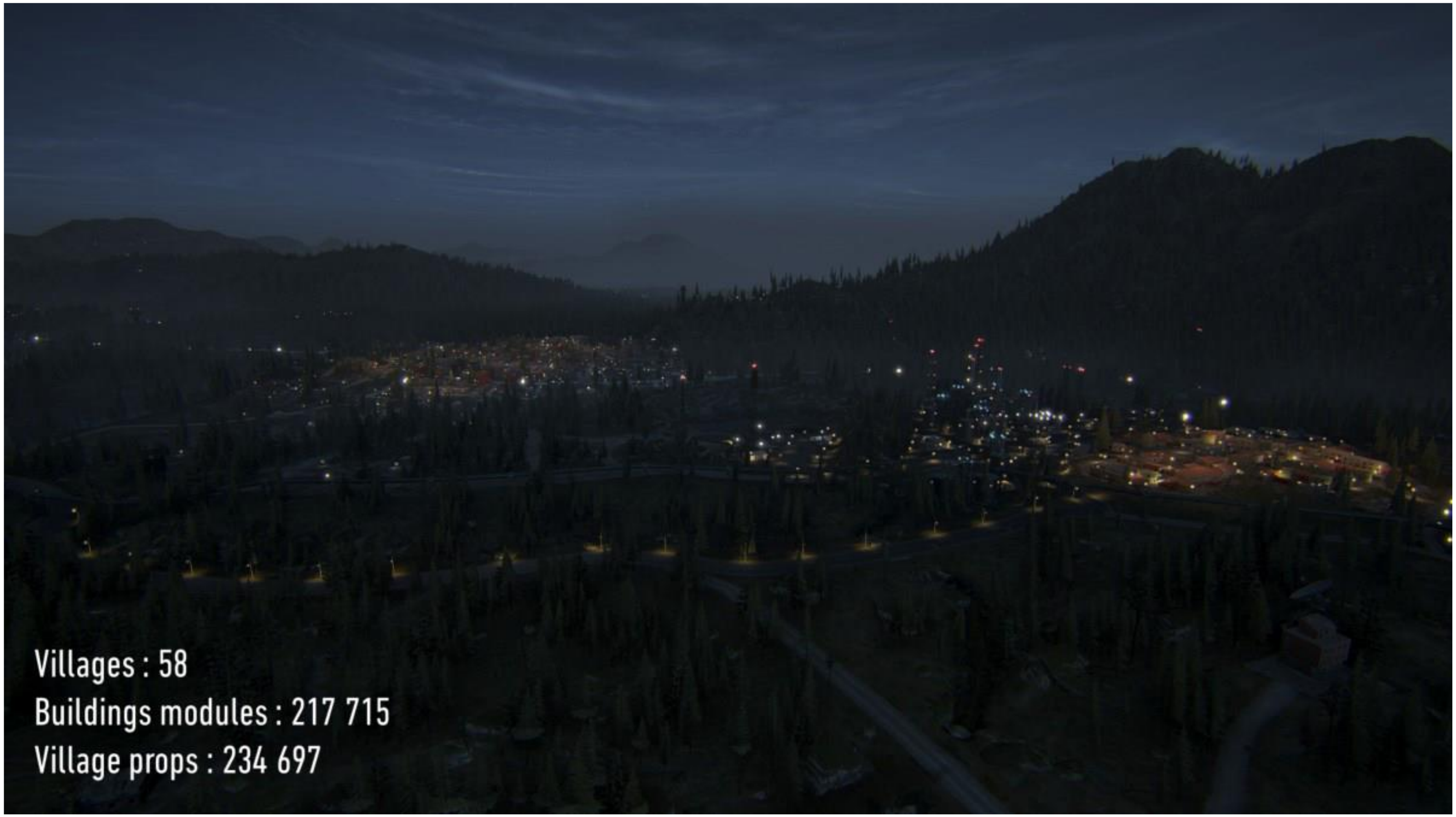
Roads : over 600km of roads and even more paths

- Railway



A full railway network going across the world

- Villages generator



Villages : 58
Buildings modules : 217 715
Village props : 234 697

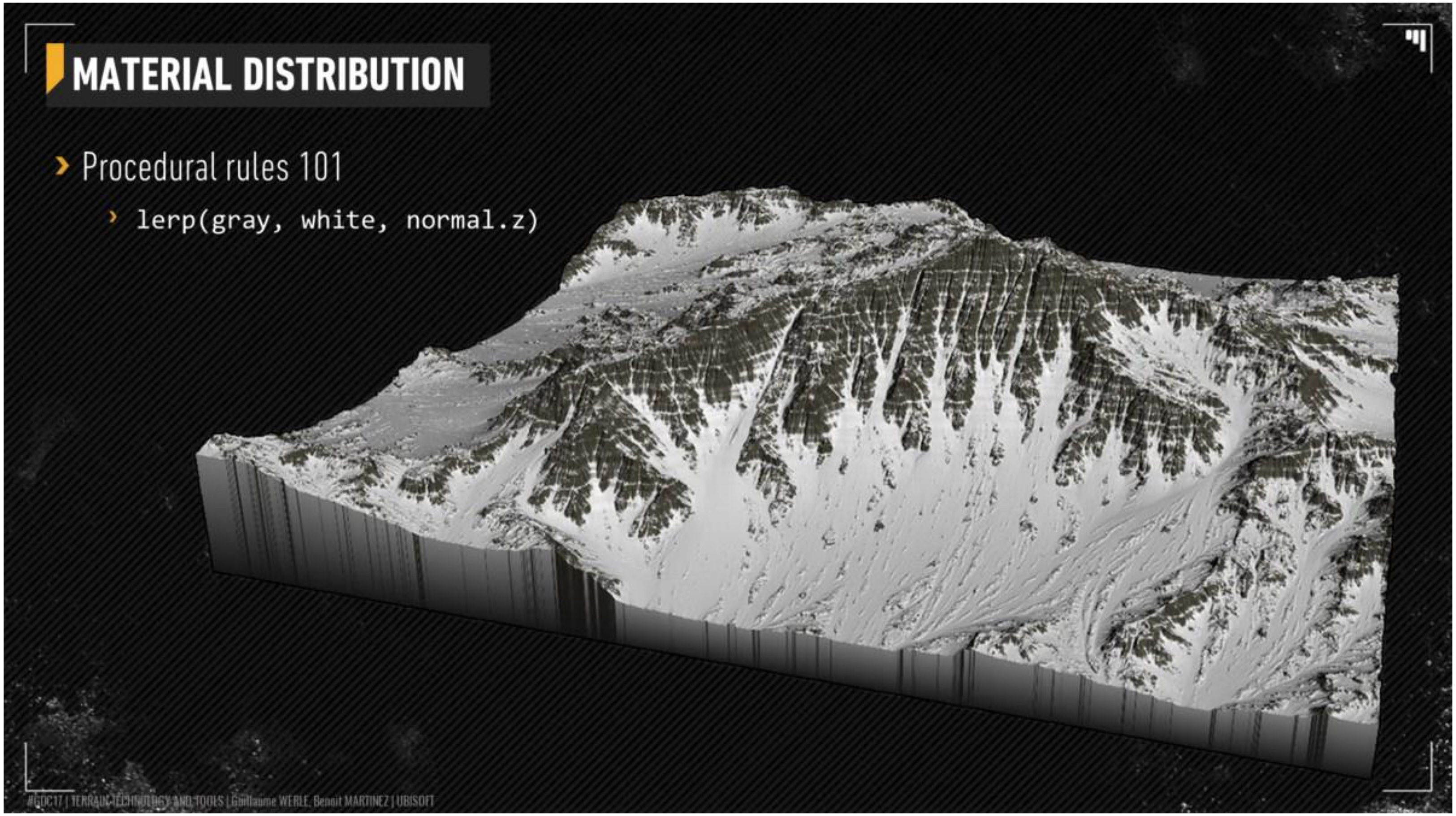
Over 200 specific locations in the game: camps, landmarks, outposts and 58 fully procedurally built villages

- Terrain modification

Base : real terrain
Editing terrain and sculpting tools; Possible to copy/ paste; Possible to erase

By looking at the size of the map led to the conclusion that painting everything manually was a terrible idea.
Need a semi automatic process.
=> Generate the material distribution procedurally

- Material distribution



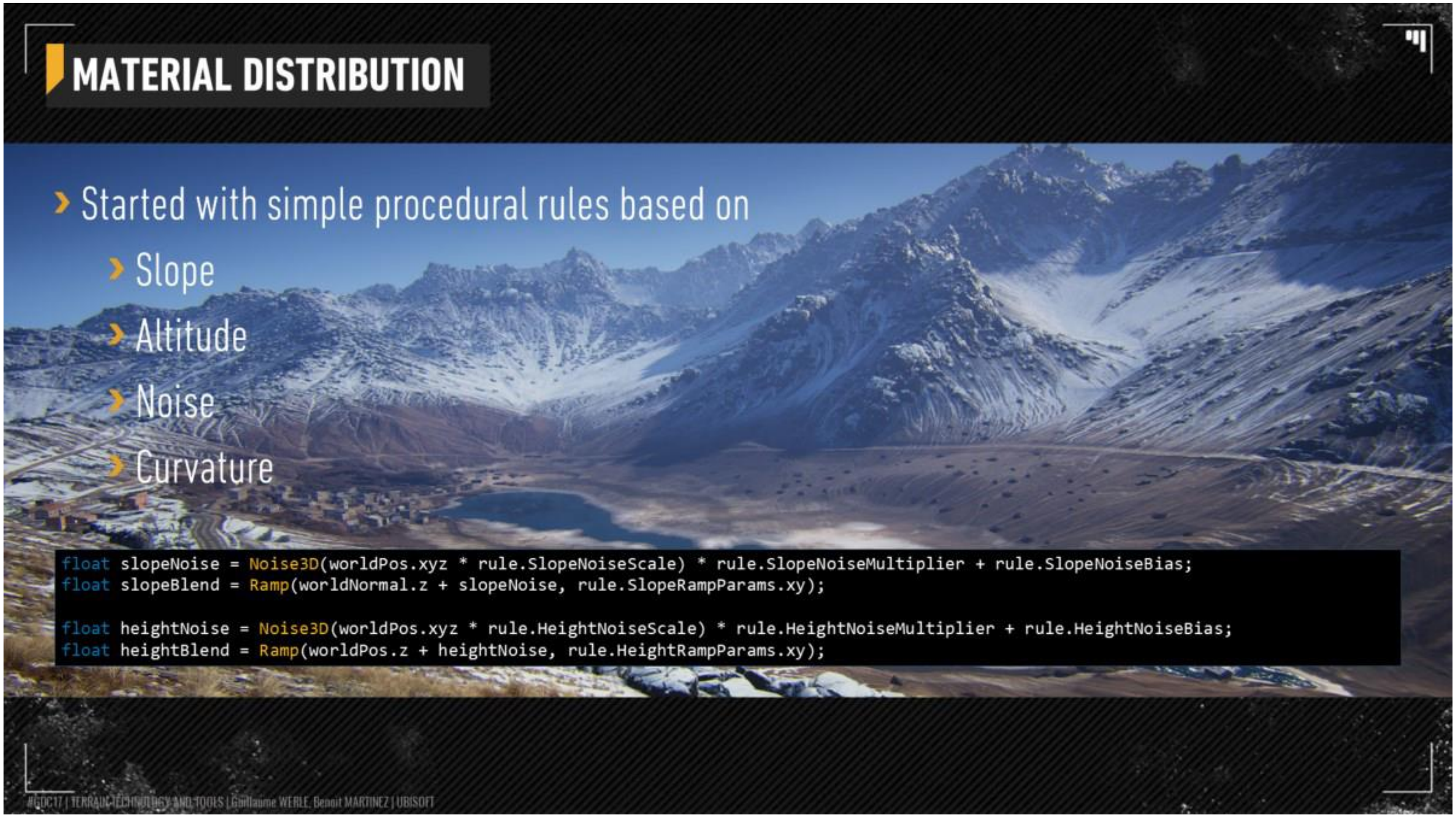
MATERIAL DISTRIBUTION

- Procedural rules 101
  - lerp(gray, white, normal.z)

#GDC17 | TERRAIN TECHNOLOGY AND TOOLS | Guillaume WERLE, Benoit MARTINEZ | UBISOFT

To give a better sense of the relief, the terrain editing tool was already tinting the heightmap depending on the normal direction.
When looking from far away, this simple rule is almost enough to generate a convincing mountain top

- Material distibution



MATERIAL DISTRIBUTION

- Started with simple procedural rules based on
  - Slope
  - Altitude
  - Noise
  - Curvature

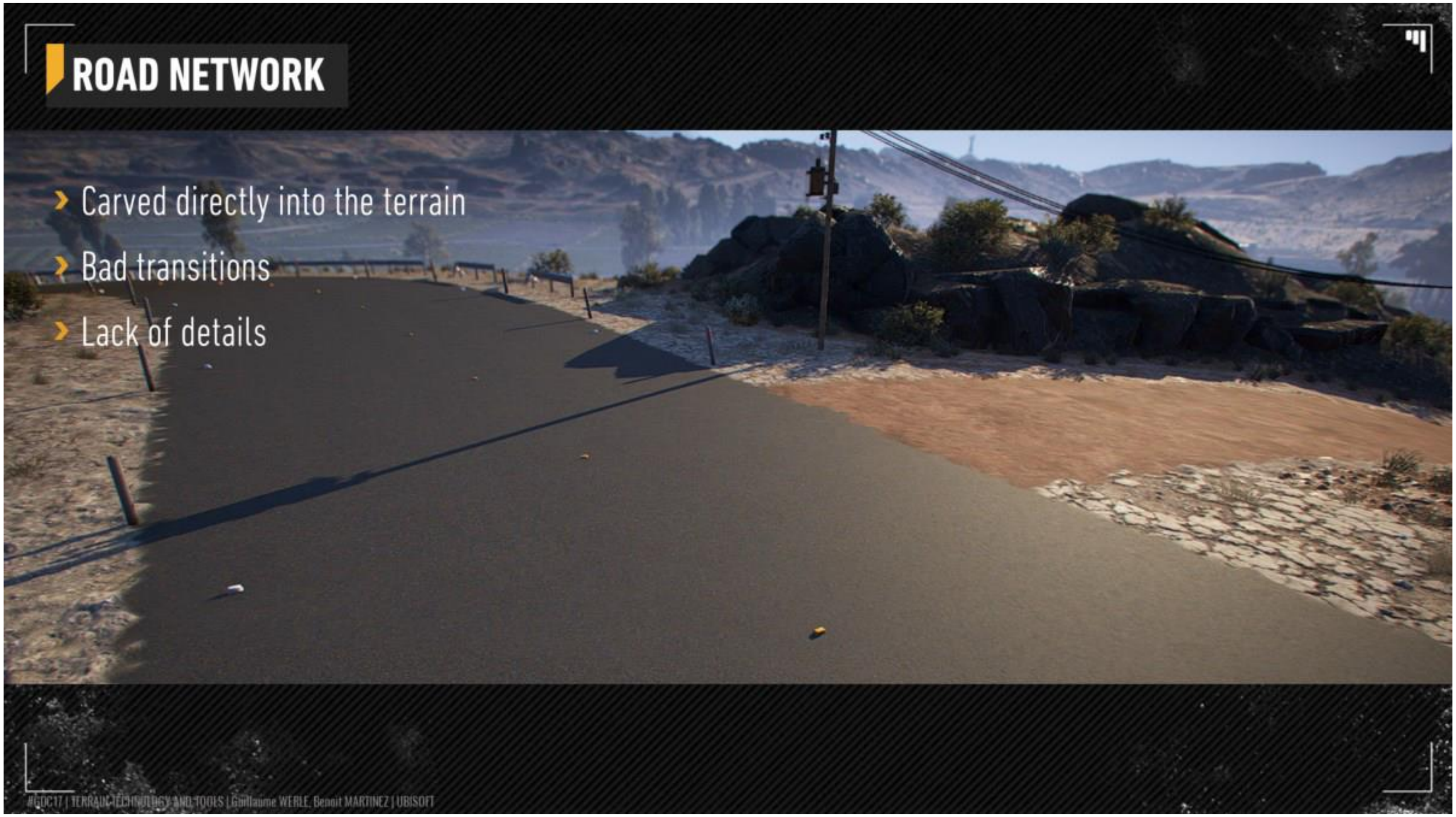
```
float slopeNoise = Noise3D(worldPos.xyz * rule.SlopeNoiseScale) * rule.SlopeNoiseMultiplier + rule.SlopeNoiseBias;
float slopeBlend = Ramp(worldNormal.z + slopeNoise, rule.SlopeRampParams.xy);

float heightNoise = Noise3D(worldPos.xyz * rule.HeightNoiseScale) * rule.HeightNoiseMultiplier + rule.HeightNoiseBias;
float heightBlend = Ramp(worldPos.z + heightNoise, rule.HeightRampParams.xy);
```

#GDC17 | TERRAIN TECHNOLOGY AND TOOLS | Guillaume WERLE, Benoit MARTINEZ | UBISOFT

Add several conditions based on the topology, noise or simple kernels that could be evaluated in real time in a pixel shader

- Road network



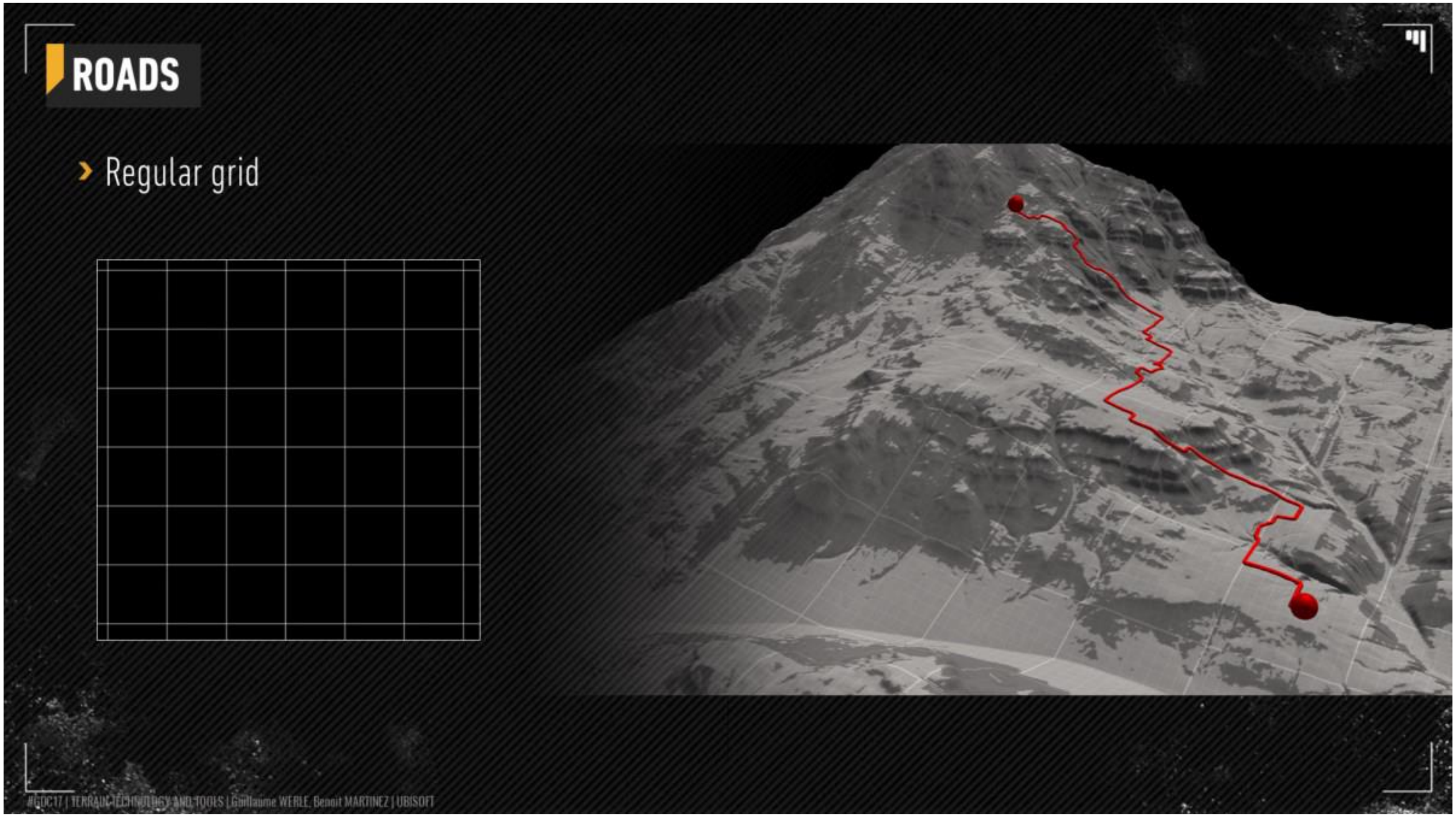
ROAD NETWORK

- Carved directly into the terrain
- Bad transitions
- Lack of details

#GDC17 | TERRAIN TECHNOLOGY AND TOOLS | Guillaume WERLE, Benoit MARTINEZ | UBISOFT

Carve/ sculpt the road directly into the terrain instead of representing it with traditional geometry.
But this technique would raised a few issue like cheap material transition and an obvious lack of details.

- Roads

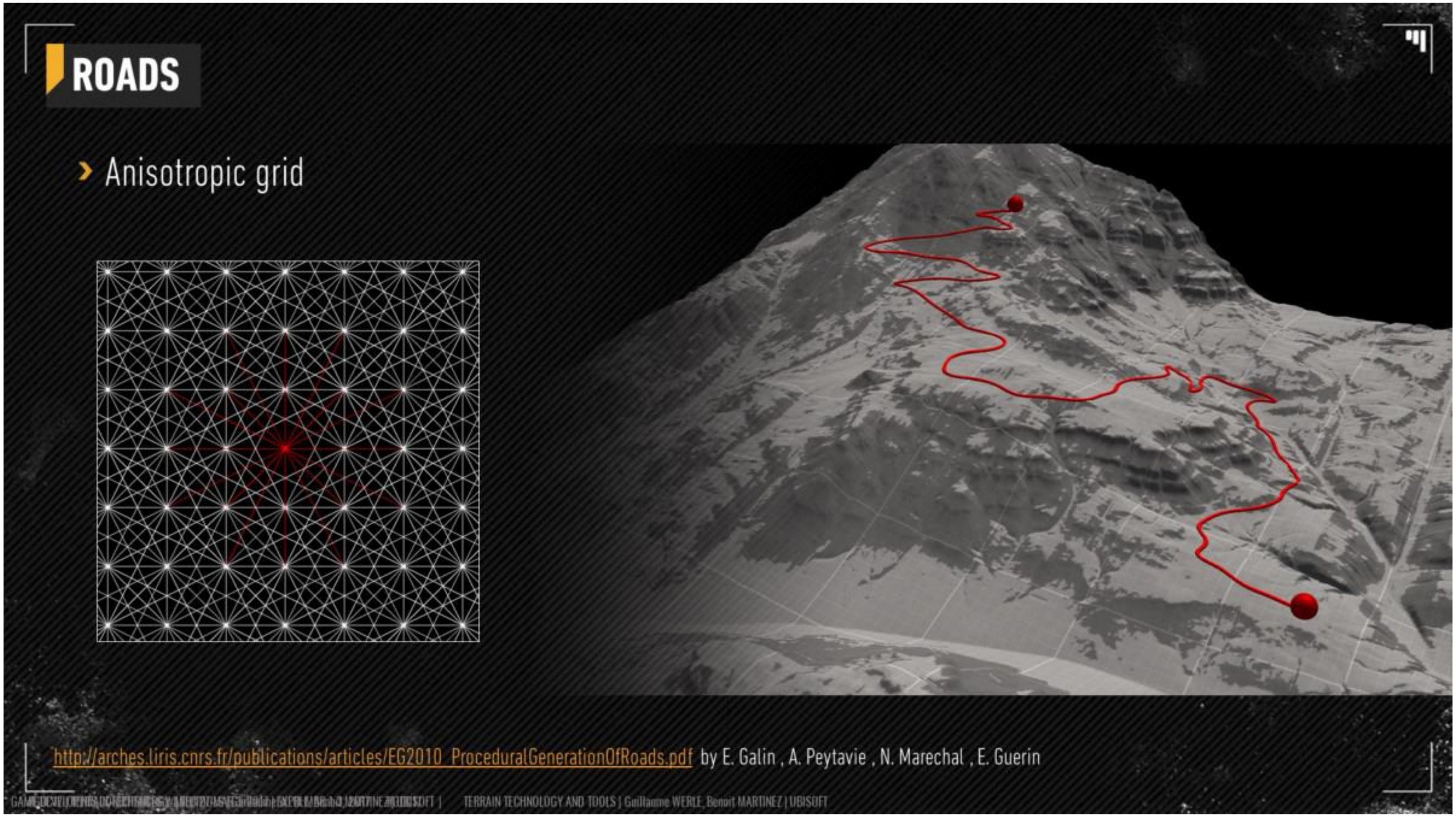


ROADS

› Regular grid

#GDC17 | TERRAIN TECHNOLOGY AND TOOLS | Guillaume WERLE, Benoit MARTINEZ | UBISOFT

- Roads



ROADS

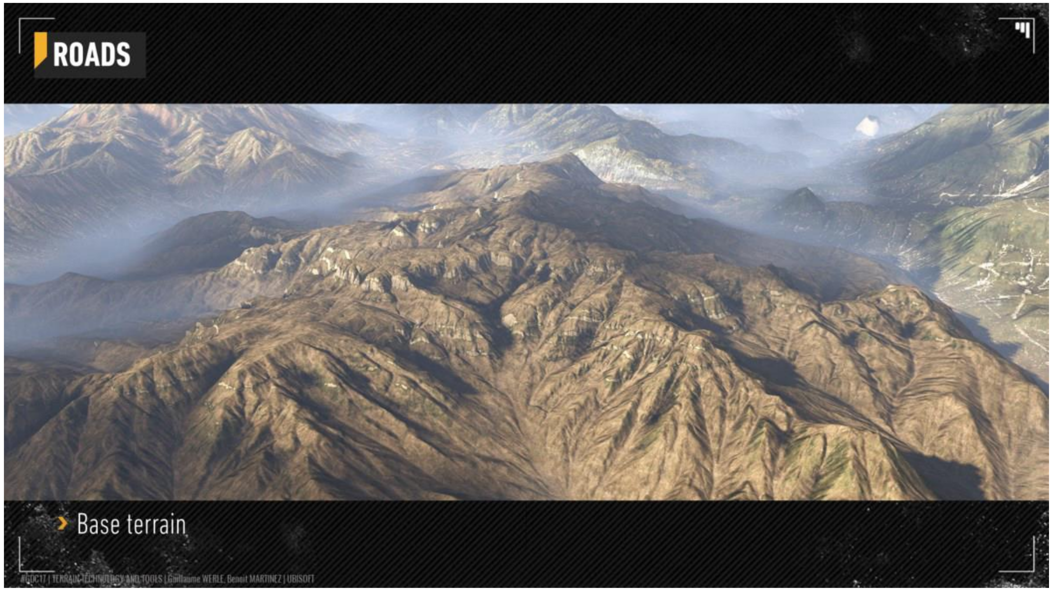
› Anisotropic grid

http://arches.liris.cnrs.fr/publications/articles/EG2010_ProceduralGenerationOfRoads.pdf by E. Galin , A. Peytavie , N. Marechal , E. Guerin

[illegible] | TERRAIN TECHNOLOGY AND TOOLS | Guillaume WERLE, Benoit MARTINEZ | UBISOFT

Using anisotropic shortest path algorithm.

- Roads network



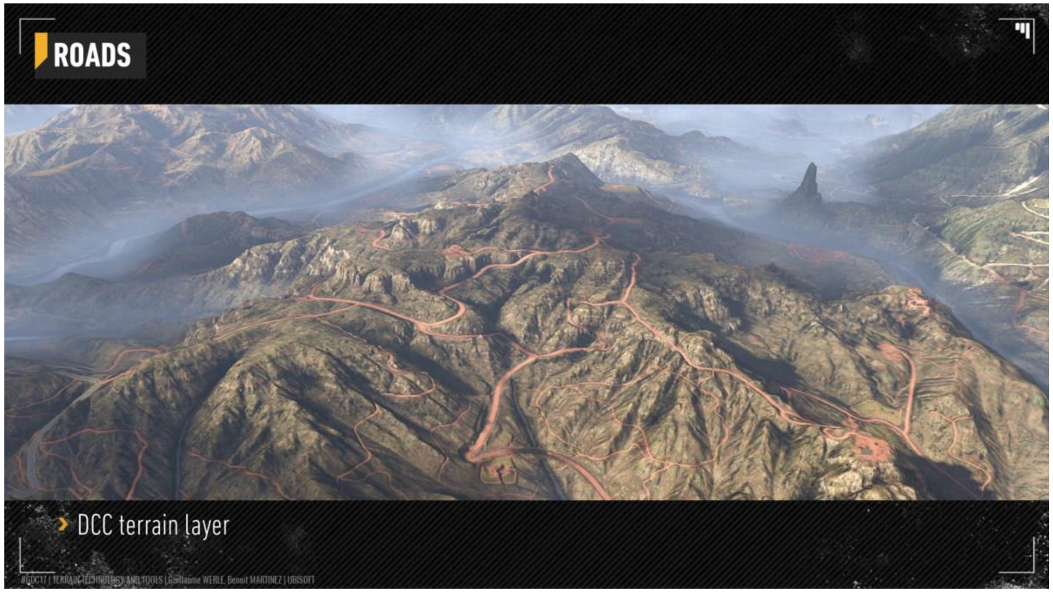
ROADS

Base terrain

#GDC17 | TERRAIN TECHNOLOGY AND TOOLS | Guillaume WERLE, Benoit MARTINEZ | UBISOFT

We obtain a consistent network with different types of roads and smallest paths.

Remember that it is all offline processed. Computing time is not (or not really) an issue (about 10minto compute a 2km*2km tile).

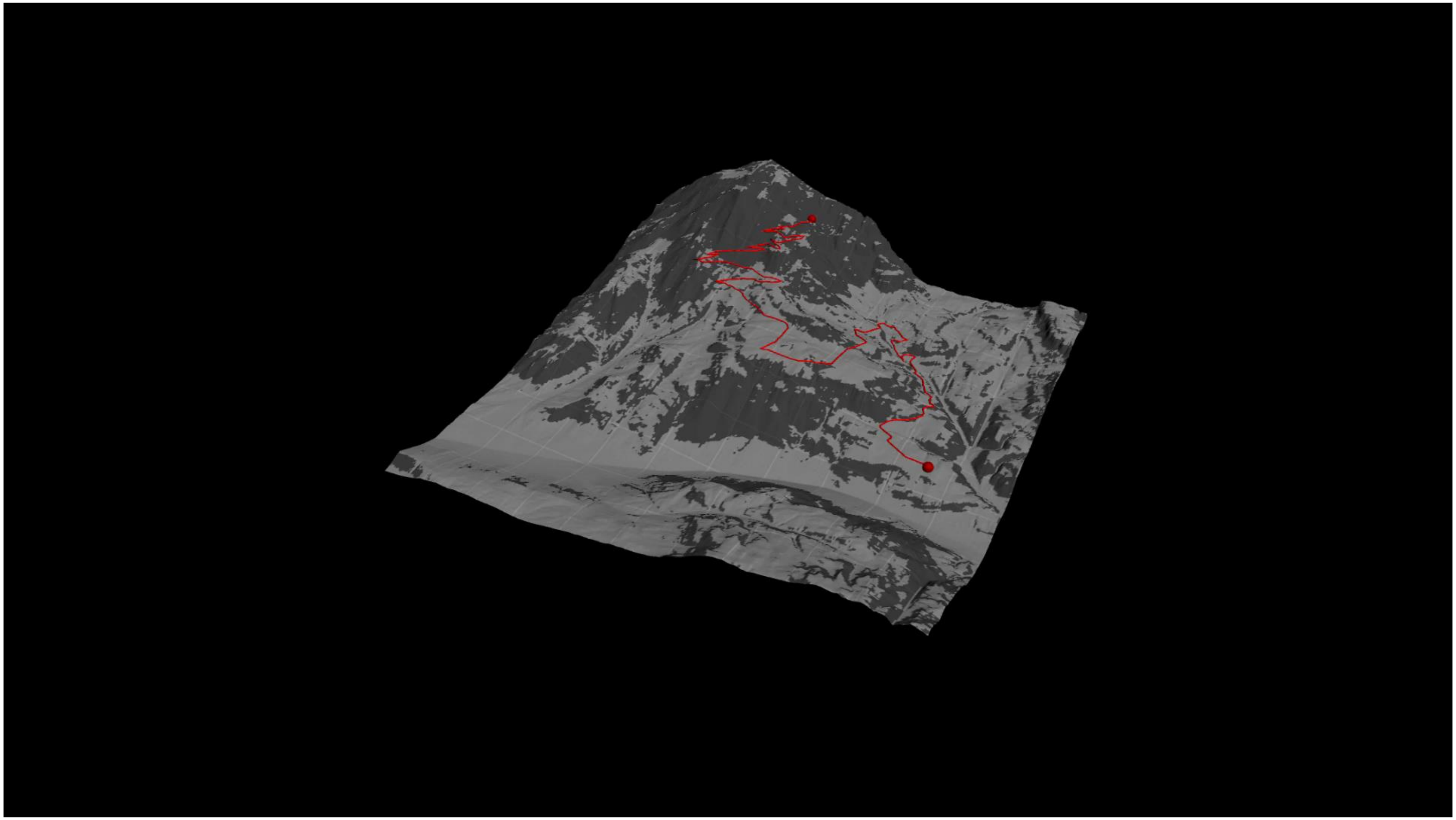


ROADS

DCC terrain layer

#GDC17 | TERRAIN TECHNOLOGY AND TOOLS | Guillaume WERLE, Benoit MARTINEZ | UBISOFT

- Roads network



Video of the pathfindingover a tile of terrain and the full extent of road network (without small paths only main roads)

- Bridges and Tunnels

https://vimeo.com/207479400

As previously mentioned some tools, like the bridge tool here, can be used in standalone mode to manually create bridges or can be embedded in a bigger tool to automatically create content.
Note: The bridges and tunnels are made of modules. No new or specific geometry is created, its all just instancing.

All the presentation of Ubisoft guys is available here
https://www.gdcvault.com/play/1024029/-Ghost-Recon-Wildlands-Terrain

- Une définition extraite de la thèse de JD Genevaux
  - La génération procédurale de contenu (Procedural Content Generation ou PCG) est une famille de méthodes où, au lieu de stocker des objets 3D, on stocke les paramètres d'un algorithme de construction. Ces méthodes ont été très utilisées dans les années 70-80, à l'époque où les ordinateurs avaient peu de mémoire. Pour créer des univers très étendus, on s'appuyait généralement sur ces méthodes qui utilisent beaucoup la notion d'aléatoire.

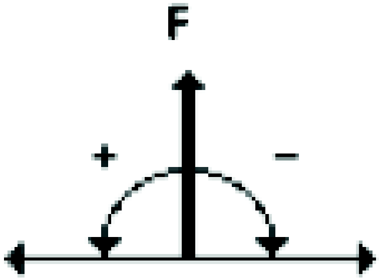
Database Amplification

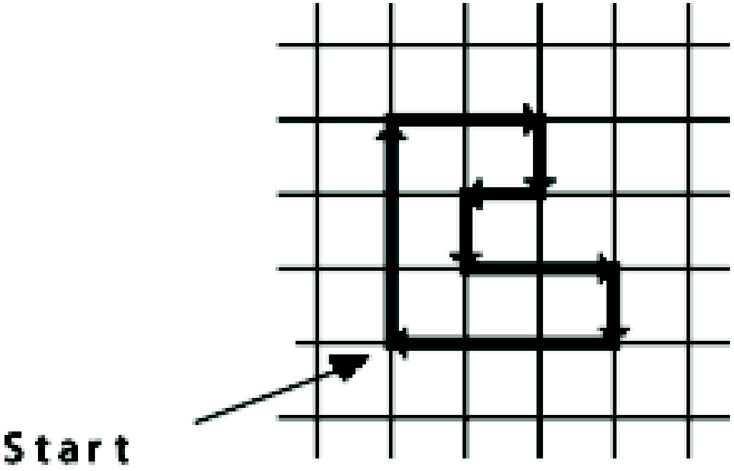
- Procedural content generation is attractive because it allows for significant database amplification
- Limited input data produces rich & varied output
  - ie: Perlin noise function + basic math gives fire, clouds, wood, etc.
- If it can be generated on the fly…
  - Level designer doesn’t have to design it
  - Don’t need to store/transmit it

L- Systems

- Developed by Aristid Lindenmayer to model the development of plants
- Excellent for modelling organic objects and fractals
- Begin with a set of "productions" (replacement rules) and a "seed" axiom
- In parallel, all matching productions are replaced with their right-hand sides
- Ex:
  - Rules:
    - B -> ACA
    - A -> B
  - Axiom: AA
  - Sequence: AA, BB, ACAACA, BCBBCB, etc.
- Strings are converted to graphic representations via interpretation as turtle graphics commands

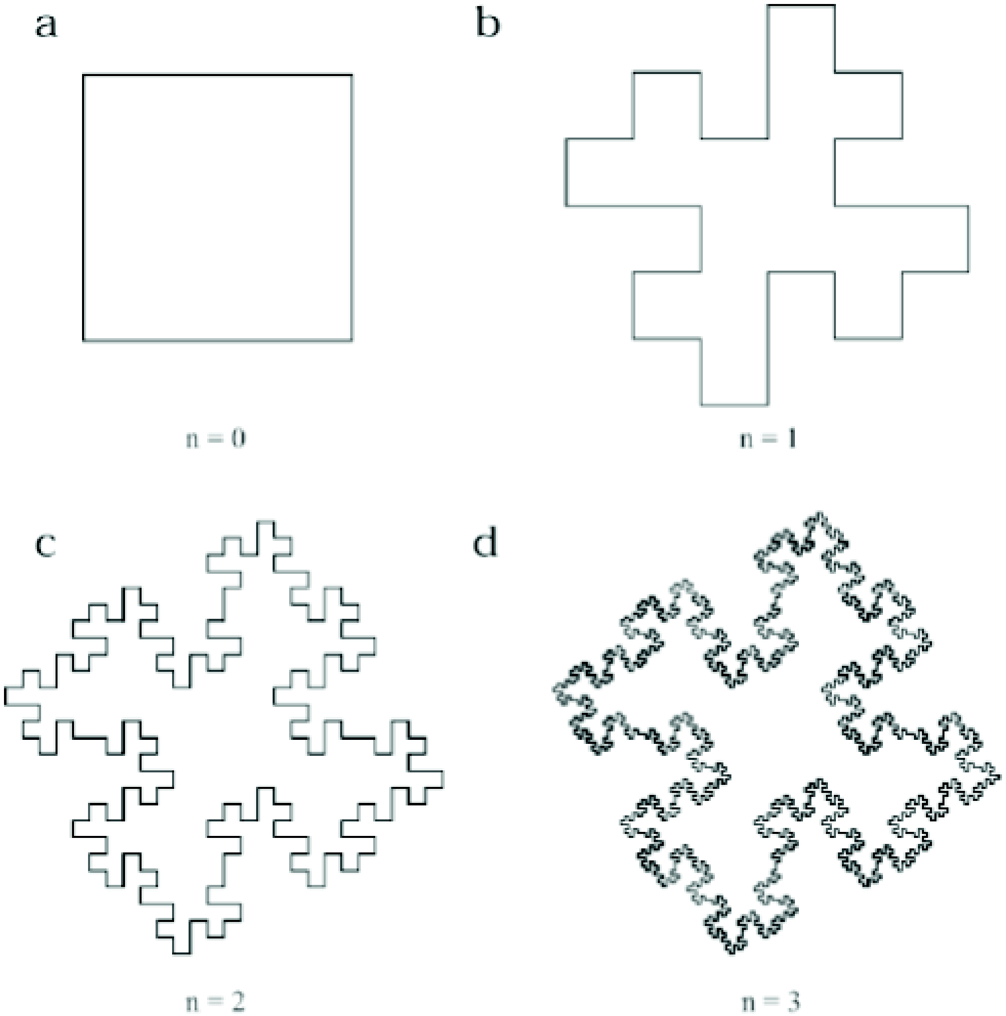
- Turtle Commands
  - F_x: move forward one step, drawing a line
  - f_x: move forward one step, without drawing a line
  - $+_x$: turn left by angle ∂
  - $-_x$: turn right by angle ∂





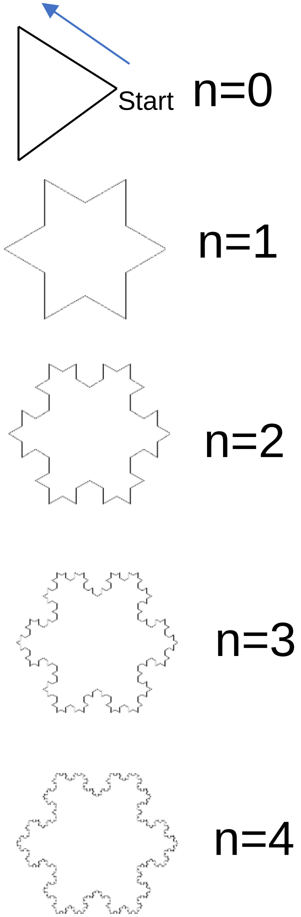
FFF-FF-F-F+F+FF-F-FFF

- Example: Koch Snowflake
  - Axiom: F-F-F-F ∂ :90 degrees
  - F -> F-F+F+FF-F-F+F

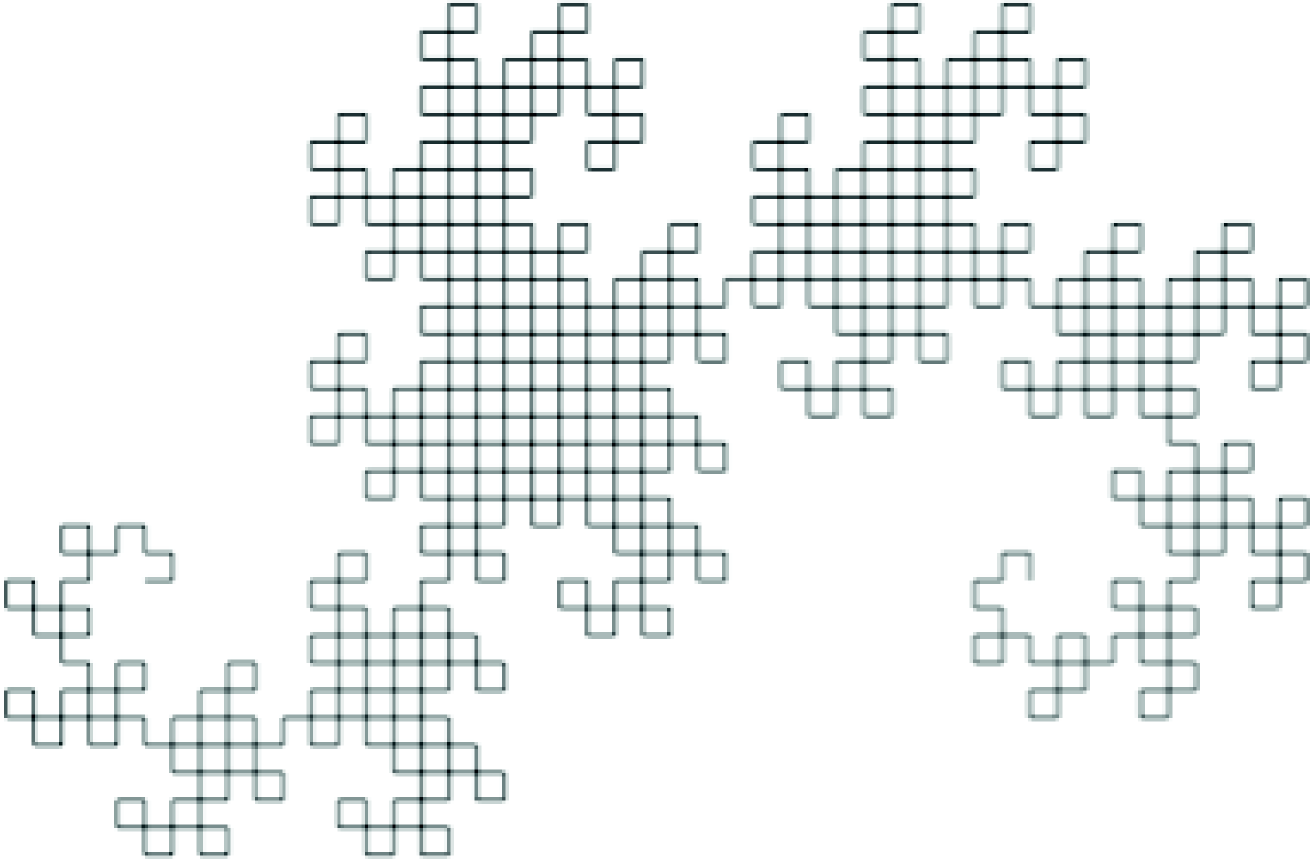


L-Systems

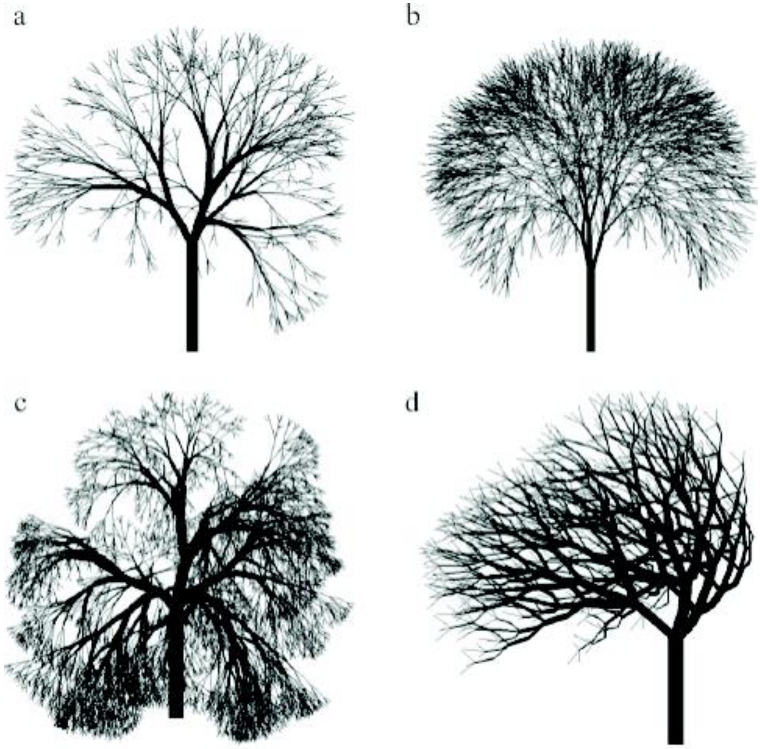
- Example : Koch Snowflakes
  - n=0 : f -- f -- f
  - n=1 : f + f -- f + f -- f + f -- f + f -- f + f -- f + f
  - n=2 : f + f -- f + f + f + f + f -- f + f -- f + f -- f + f + f
    + f -- f + f -- f + f -- f + f + f + f + f -- f + f -- f + f -- f
    + f + f + f -- f + f -- f + f -- f + f + f + f + f -- f + f -- f
    + f -- f + f + f + f -- f + f
  - Assign meaning to each component of the alphabet
    - f draw forwards
    - + turn 60 degrees right
    - -- turn 60 degrees left



- Example: Dragon Curve
  - Axiom:F_l ∂ :90 degrees n:10 iterations
  - F_l -> F_l+F_r+
  - F_r -> F_l-F_r-



- Grammar: Extensions
  - Basic L-Systems have inspired a large number of variations
  - Context sensitive: productions look at neighboring symbols
  - Bracketed: save/restore state (for branches)
  - Stochastic: choose one of n matching productions randomly
  - Parametric: variables can be passed between productions



P. Prusiniewicz

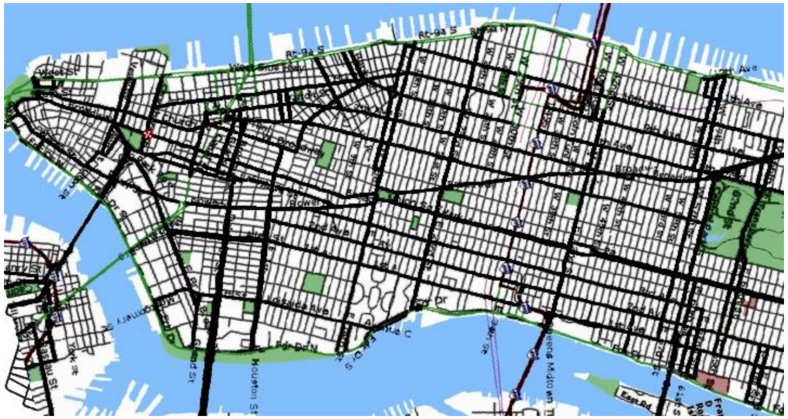
- Others links
  - Algorithmic Botany
    - Covers many variants of L-Systems, formal derivations, and exhaustive coverage of different plant types.
    - http://algorithmicbotany.org/papers
  - PovTree
    - http://arbaro.sourceforge.net/
  - Trees construction : Soeren Pirk :
    - https://www.youtube.com/watch?v=5HJSHJvlMFE

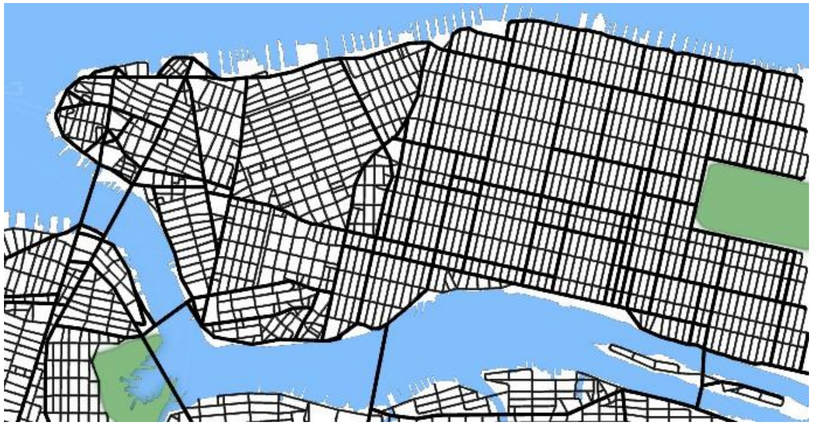
- Exemples

https://www.youtube.com/watch?v=E1B4UoSQMFw&list=PLQu8TxiWYLxID80vTywVWjaloCyytE2gB&index=2

L-Systems for Cities [Parish01]

- Cities generator





  - Start with a single street
  - Branch & extend w/ parametric L-System
  - Parameters of the string are tweaked by goals/constraints
  - Goals control street direction, spacing
  - Contraints allow for parks, bridges, road loops
  - Once we have streets, we can form buildings with another L-System
  - Building shapes are represented as CSG operations on simple shapes

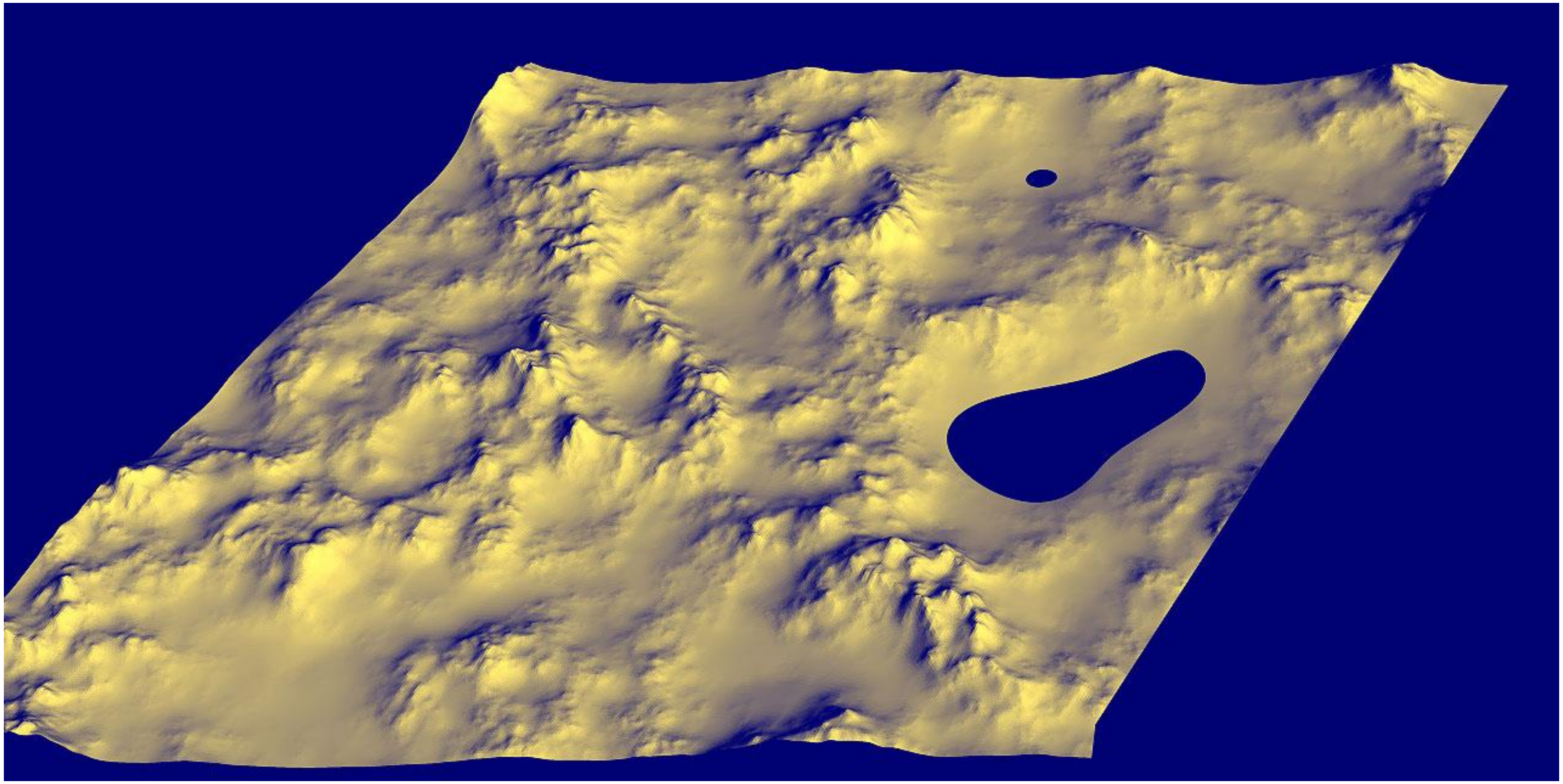
Procedural Modeling of Cities, Parish, Müller, 2001
https://cgl.ethz.ch/Downloads/Publications/Papers/2001/p_Par01.pdf)

- Noise Functions
  - Seeded pseudo-random number generator
  - Over R^n
  - Approximation to gaussian filtered noise
  - Implemented as a pseudo-random spline
  - The trick is to make it fast

https://www.youtube.com/watch?v=wbpMiKiSKm8

- Using Perlin Noise in Unity
  - Noise : https://www.youtube.com/watch?v=WP-Bm65Q-1Y
  - Parameters :
    - Width / Height : taille de votre texture
    - Scale : l'echelle de votre texture
    - Octaves : le nombre d'appels à la fonction de de bruit
    - Persistance : correspond au facteur venant modifier l'amplitude de chaque fonction de bruit
    - Lacunarity : la fréquence entre chaque octave

- Adding water
  - Use an elevation threshold ($z < z_{water}$)



F.K. Musgrave

Some links

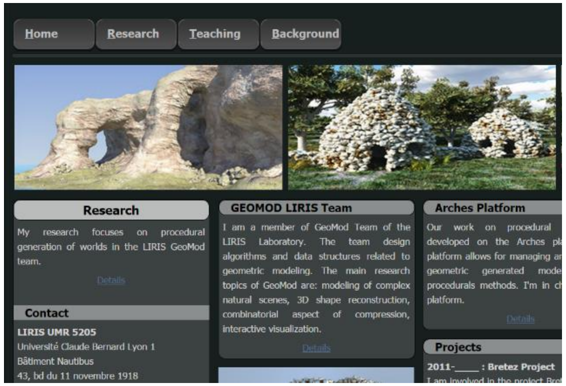
- http://martindevans.me/game-development/2015/12/11/Procedural-Generation-For-Dummies-Roads/
  - Road (with grid or randomized)
  - Buildings (differents buildings)
  - Vegetation with L-System
  - Terrain generation

Some links

- https://www.youtube.com/watch?v=Jsc3BQaJndQ (procedural generation of city in Unity)
- https://www.youtube.com/watch?v=xGi0Hr-kNw0 (Procedurally Generated City And Creating A Grid System)
- https://www.youtube.com/watch?v=xkuniXI3SEE (Generating a Procedural City with Unity 5 Part 1)
- https://marian42.de/article/wfc/ (Infinite procedurally generated city with the Wave Function Collapse algorithm)

Some links

- Génération d’objets à partir d’observation du monde réel…



Research

My research focuses on procedural generation of worlds in the LIRIS GeoMod team.

Details

Contact

LIRIS UMR 5205
Université Claude Bernard Lyon 1
Bâtiment Nautibus
43, bd du 11 novembre 1918

GEOMOD LIRIS Team

I am a member of GeoMod Team of the LIRIS Laboratory. The team design algorithms and data structures related to geometric modeling. The main research topics of GeoMod are: modeling of complex natural scenes, 3D shape reconstruction, combinatorial aspect of compression, interactive visualization.

Details

Arches Platform

Our work on procedural
developed on the Arches pla
platform allows for managing ar
geometric generated mode
procedurals methods. I'm in ch
platform.

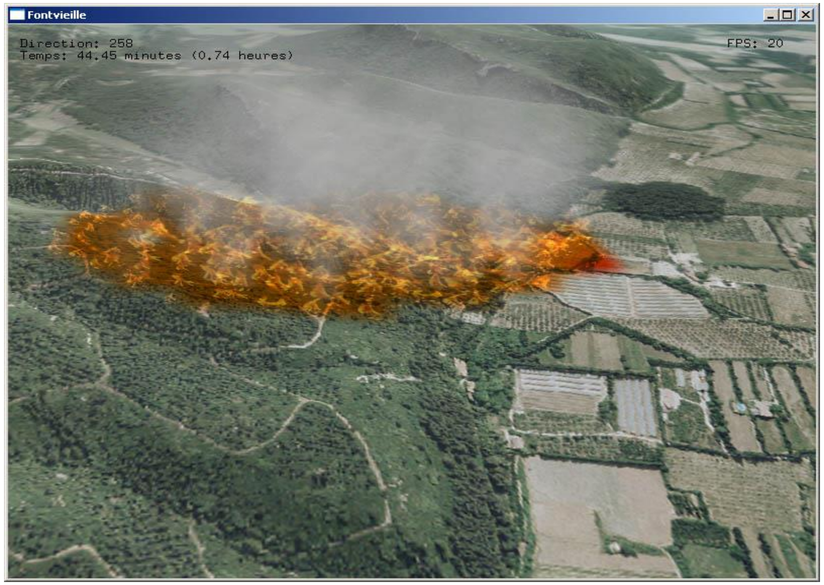
Details

Projects

2011-____ : Bretez Project

http://arches.liris.cnrs.fr/

- Génération avec des données réelles



Fontvieille

```
Direction: 258                                    FPS: 20
Temps: 44.45 minutes (0.74 heures)
```

GRANDLYON
communauté urbaine