On Modular Assets and Substances in Unity

PROCEDURAL ART

2021-2022

LECTURE 3 – Art- & Design oriented

by Max Klostermann

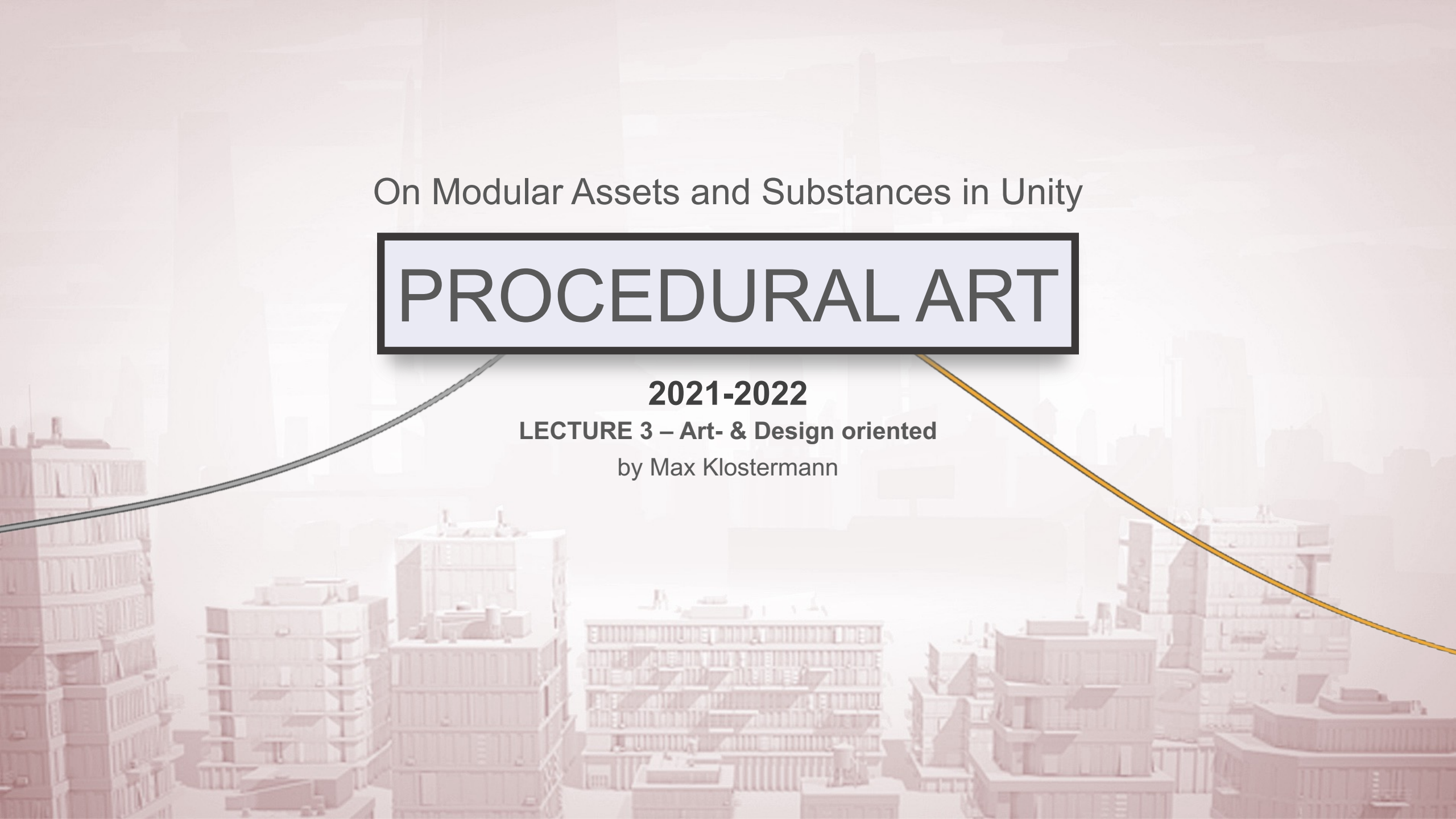


TABLE OF CONTENTS

THE IDEA BEHIND MODULARITY

Let's create a poll:

"The **biggest strengths** of **modular assets** are…."

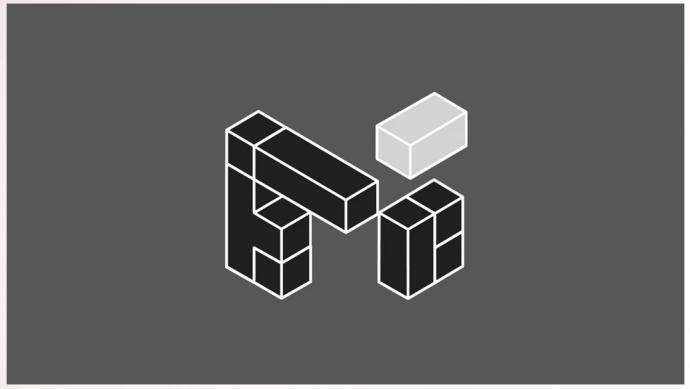
A. its reusability
- reusing sections & elements elsewhere to save production time

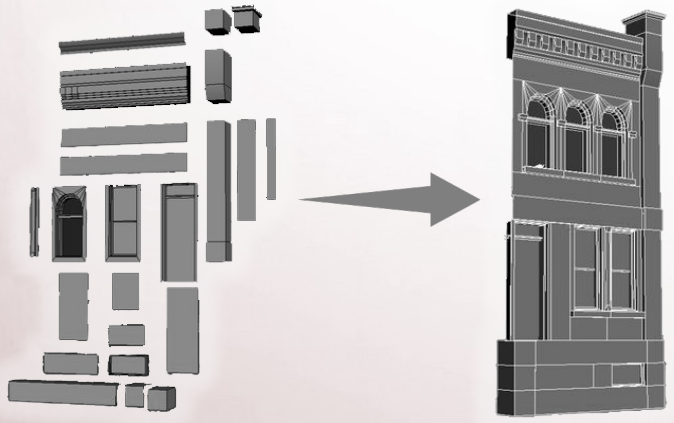
B. its adaptability
- making changes to elements which affects the elements elsewhere

C. its visuals
- the unique flavor of all individual elements & reusing those

D. its efficiency
- reusing sections & elements elsewhere to save performance

THE IDEA BEHIND MODULARITY





About Modularity

- found in disciplines such as architecture, technology, biology and most importantly for us: **VFX and game development**
- modularity is about:
  - *"**assembling** a number of fixture **elements in a** feasible **sequence** that is properly **determined**."*

Uday Hameed Farhan, 2013

- for VFX and game development it is applied through
  - *"the creation of **smaller components** designed with the intent of being **combined for** the **creation of new assets**, **props** and **structures**."*

Braiden Fenech & Justin T. Carter, 2017

THE IDEA BEHIND MODULARITY

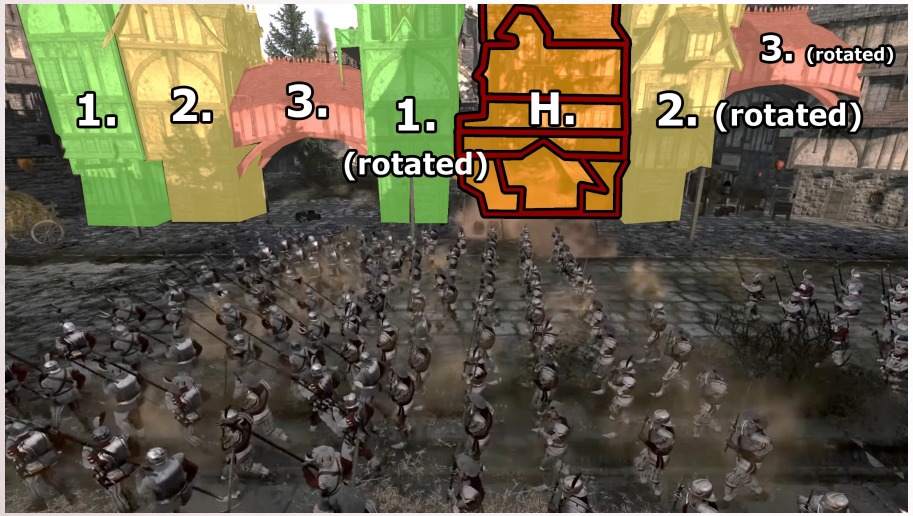
Modularity in games

1. contemporary **example**:

   Dark Souls III (FromSoftware, 2016)

   - uses a **modular design** for its 3D environments **to**
     - **create all** architecture found within the game
     - **increase flexibility** and **reduce** production **time**
   - uses mostly **planar** models that show **details through materials** (roughness, normal and height maps)
   - uses only very **few unique** models
   - uses minor **model-** and **material variations** and **foliage** to **hide** the **repetitious** nature of its modular assets

THE IDEA BEHIND MODULARITY



Modularity in games

2. contemporary **example**:

   Total War: Warhammer (The Creative Assembly, 2016)

   - uses a **modular design** for its 3D environments **to**
     - **create** all its architecture with simple models
     - give **just enough variety** through different building types
   - uses mostly simplistic **planar** modeling and **primitives**
   - many assets are **reused** and simply **rotated** (90° - 270°)
   - some ‘hero’ houses (H) consist of numerous **prefabricated floors** which are **stacked**, and **rotated** randomly to **break** up **repetitiveness**

THE IDEA BEHIND MODULARITY

Main benefits of modularity

▲ **Reusability:** Artists **don't** need to **create** a **thousand** individual **unique** looking **assets** (one set of modular assets can theoretically be enough to populate an entire village)

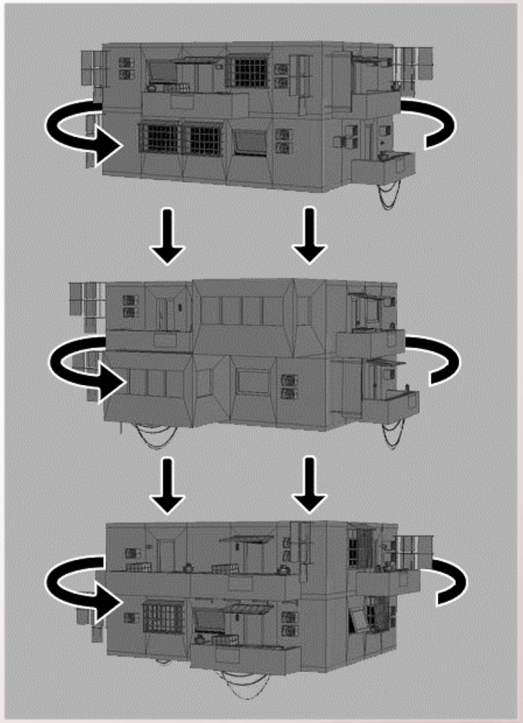
▲ **Efficiency: The more** elements are **reused**, **the less** computation **time** the **CPU** will take (more **F**rames **P**er **S**econd due to fewer draw calls, therefore, less to do for the CPU)

Potential benefits of modularity

► **Adaptability:** Changing models/materials (**prefabs**) and thus, building blocks changes the same assets elsewhere (this **can** be beneficial, but **might** also **hinder** and **prolong production time**)

Drawbacks of modularity

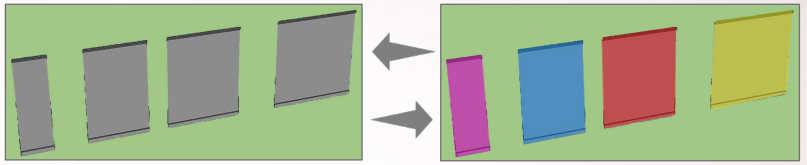
▼ **Visual fidelity: Reused assets** make it **harder** to **maintain** a believable **visual diversity** (the human **eye easily detects repetition**; **actions** to circumvent this often lead to **increased costs** by having to develop and produce unique **assets** and **shaders** that 'cover up' this repetition)



COMMON METHODS

1. **Planar** method / modularity
   - designers use **one-sided uniform modules** which are **snapped together** in a ‘jigsaw’ style

– **Main observations**
   - **Primary assets to develop:**
     - wall modules, pillars, cornices, trim modules, floor modules, ceiling modules and similar
   - **Specific benefits and/or drawbacks:**
     - best for interior or smaller environments done manually (or through elaborate scripts/tooling)
     - production time high when done through manual labor



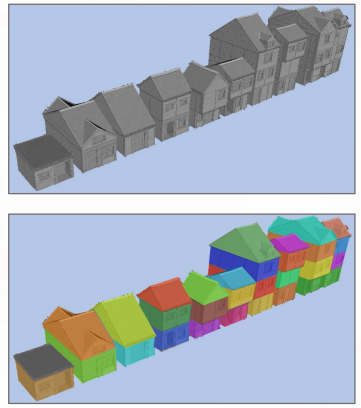
World of Darkness MMO (2013)

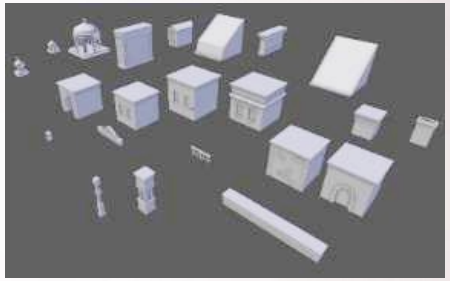
XCOM 2 (2016)

COMMON METHODS

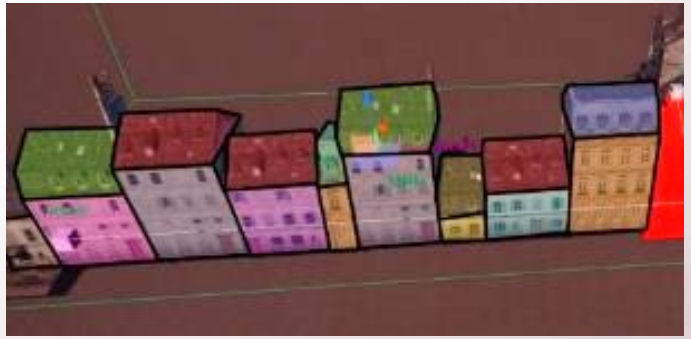
2. **Box** method / modularity
   - designers incorporate **larger intersecting assets** with **each** individual **asset** featuring a **higher level** of **detail** (or illusion of **complexity**, reducing repetitiveness as much as possible)

– **Main observations**
   - **Primary assets to develop:**
     - Square building modules, roofing modules, pillar modules and similar
   - **Specific benefits and/or drawbacks:**
     - best for large outdoor environments where intricate detail is not necessarily needed
     - not always fit for optimization due to potential rendering of unseen faces, less culling & LOD transitions





Individual Box Modules

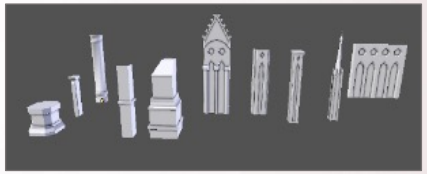


Assassins Creed Unity (2014)

COMMON METHODS

3. **Diverse** method / modularity
   - designers develop a **series** of **non-uniform modules** which are designed to **interact with** every **other module** in an **unconventional** manner

– **Main observations**
   - **Primary assets to develop:**
     - roof modules, window modules, wall modules, pillar modules and similar (all in multiple variations)
   - **Specific benefits and/or drawbacks:**
     - provides most visual fidelity and artistic freedom
     - best suited for manually crafted, linear experiences to better optimize the scene due to detail density of assets
     - time-consuming asset creation prior to implementation into the game engine



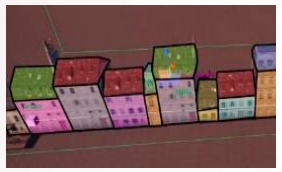
Assets and structures from Bloodborne (2015)

Modular Building from The Sinking City (2019)

DIFFERENT APPROACHES

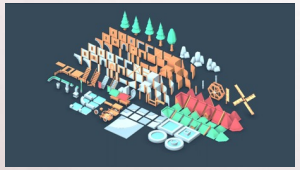
What method to use for your city area?

1. **Planar**: When you have **or** create a script that **snaps** individual **planes** and **stacks them** in a “controlled-random” order, go for it (editor tools preferable).



2. **Box**: Perhaps the ‘easiest’ method to use. You prepare **stackable** building **blocks** (floors & roof tops) and have more artistic freedom with each. Your script should allow for **stacking** and **editor tooling** for further control (alternative: manual stacking).

3. **Diverse**: Here there are no limitations. It’s a **mix** between **planar-** and **box modularity** but will involve the **most amount** of detailed **modelling** (not the focus of this course). Potentially also the **least amount** of procedural **control** and **more manual labor**.

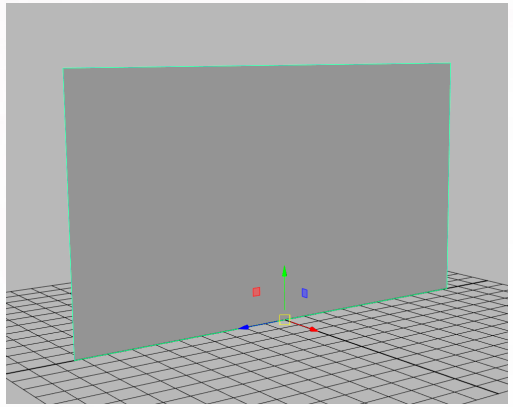


4. **Replacement**: By replacing the **Kenney Prefab Assets** with your **own meshes** and **materials**, you have one script already working for you. Now ask for support to make it do more (or adapt the buildings manually). Do also checkout the script-oriented lectures!

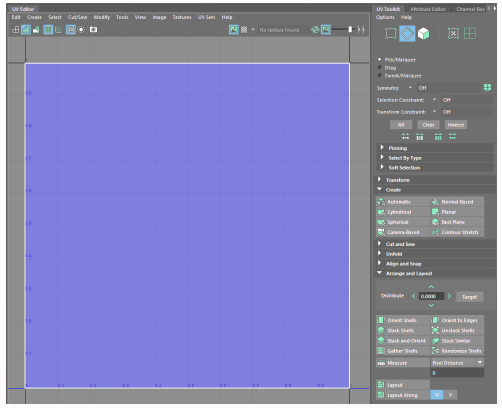
On Modular Assets

PLANE AND SIMPLE

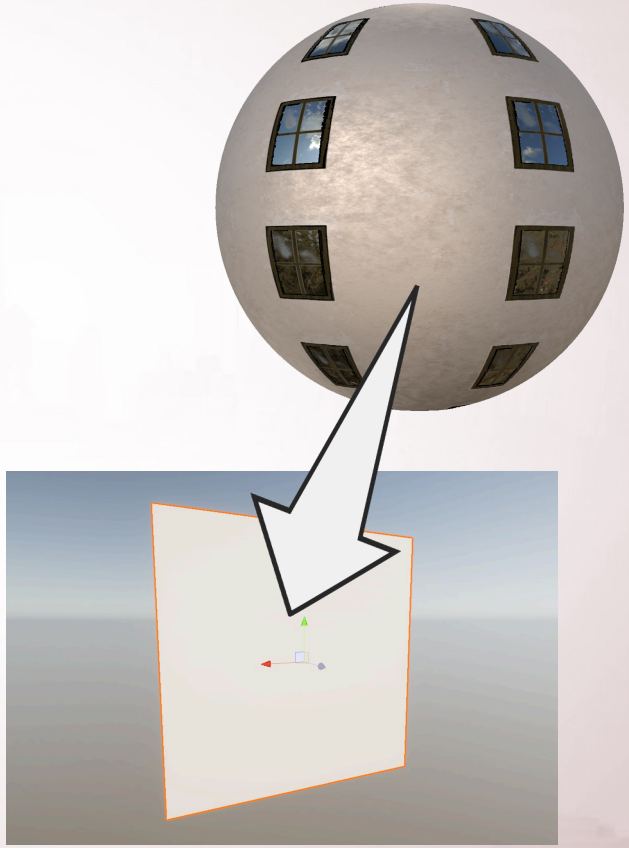
Tip for all methods:
use properly tiled **substances** with visible **normal & height** map **detail** (window frames, doors, wooden beams, "cyber" panels, etc.) on most, if not all modular assets



Pivot points: Make sure the pivot is set correctly for each modular asset (depends on the way stacks and floors are aligned and snap together).



UVs: Ensure that the UVs for planar meshes (and possibly all others too) are using the entire 0 to 1 space on your UV layout. Do tiling value adjustments inside of Unity if necessary.



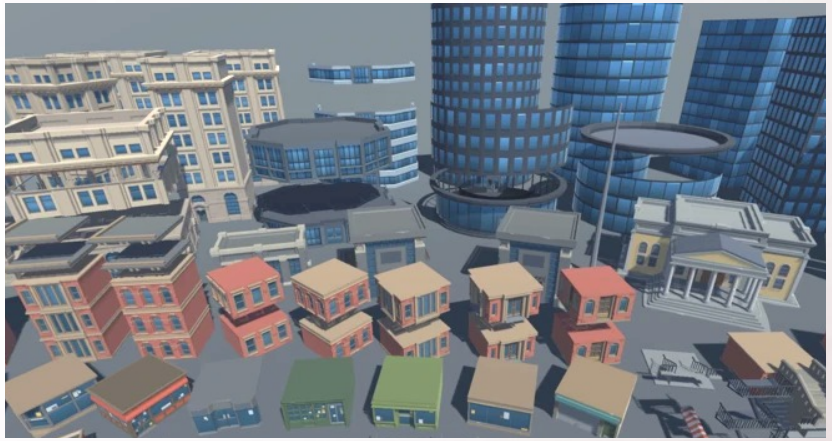
In Unity: Drag & Drop the Substance Graph and adjust the material settings.

On Modular Assets

MAKING THE CITY AREA HAPPEN

First things first:

1. Don’t panic.
2. We **don’t expect** a full-blown **cityscape**.
3. We **expect** a **reimagined version** of a small city area based on the game of your choosing.
4. It **can be low-poly** with **materials** that **bring back** the **missing** geometrical **detail**.
5. It is about **capturing** a “**skyline impression**” (recognizing the city from a distance, **not** up close)



Much closer to what we expect than all those references.
Lots of room for better materials here though.

On Modular Assets

MAKING THE CITY AREA HAPPEN

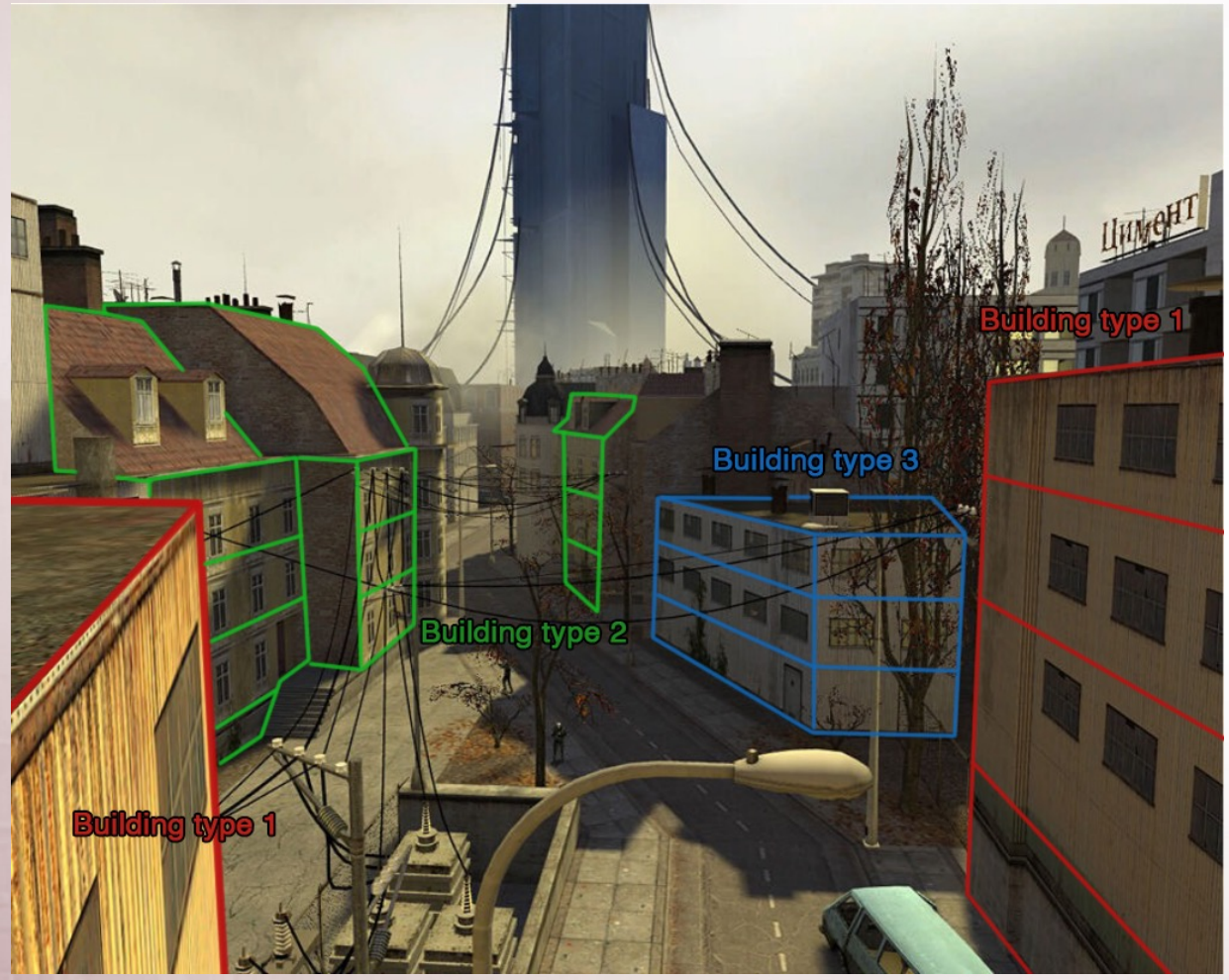
Breaking it off into comprehensible chunks

City 17 – City Center (**Half-Life 2**)

Tips for getting it done:
- **don't zoom in** extremely
- create a **color palette** with the most **dominant colors**
- analyze and **boil it down** to 4-5 reoccurring floor types (+ roofs) and the cities' main materials
- the **shapes** can be quite **low-poly** and get the lost geometrical detail back through substances with details (windows, rust, plaster color etc.)

On Modular Assets

MAKING THE CITY AREA HAPPEN



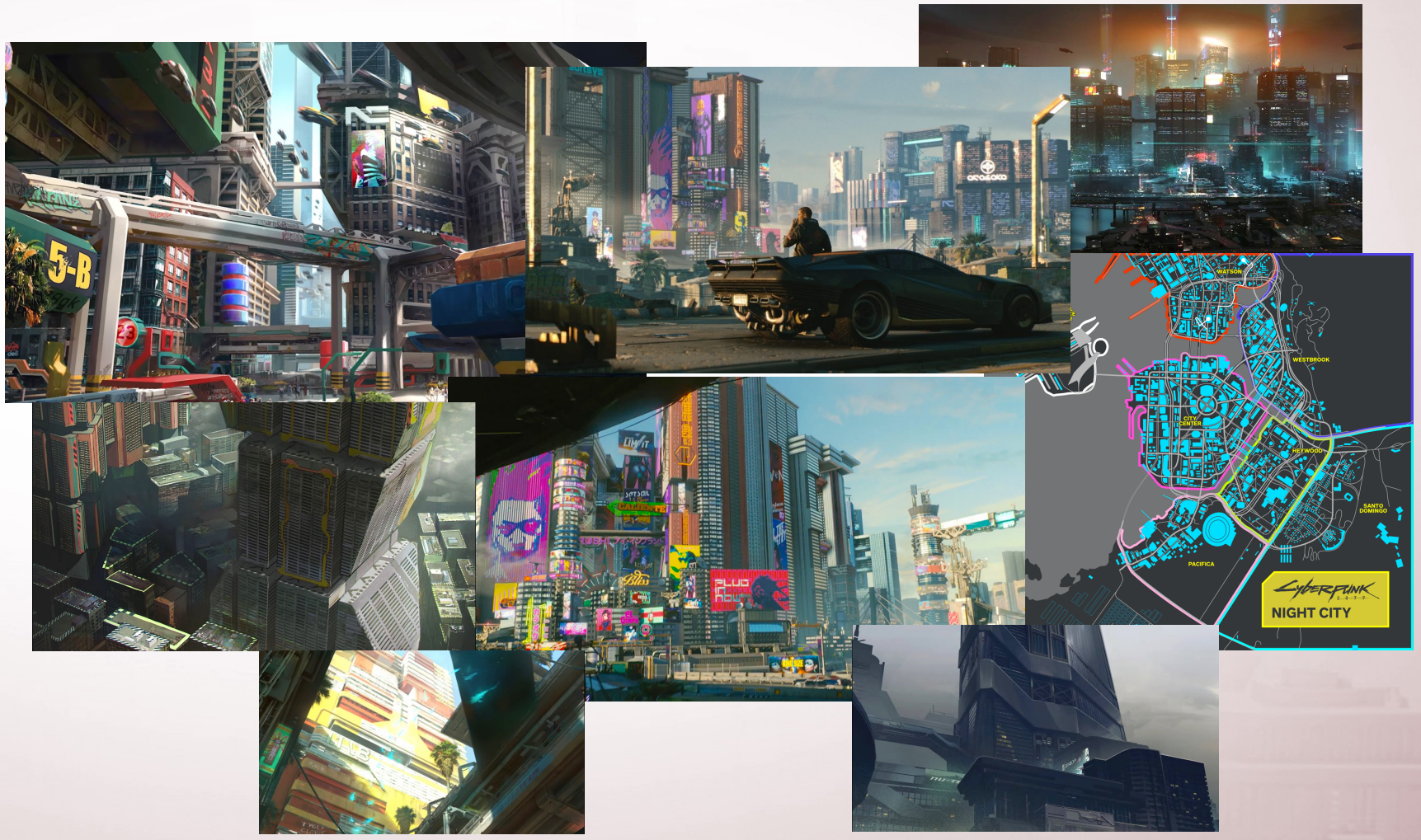
On Modular Assets

MAKING THE CITY AREA HAPPEN

Breaking it off into comprehensible chunks

Downtown Night City **(Cyberpunk 2077)**

Tips for getting it done:
- **don't zoom in** extremely
- create a **color palette** with the most **dominant colors**
- analyze and **boil it down** to 4-5 reoccurring floor types (+ roofs) and the cities' main materials
- the **shapes** can be quite **low-poly** and get the lost geometrical detail back through substances with details (windows, rust, plaster color etc.)



On Modular Assets

MAKING THE CITY AREA HAPPEN

Look for **reoccurring patterns** and recreate them using height information via Substances; adjust tiling

Substance fundamentals

A WALL WITH WINDOWS

Quickly creating a wall with some simple

window in Substance 3D Designer

Link to Blackboard's substance resources page:
https://leren.saxion.nl/webapps/blackboard/content/listContent.jsp?course_id=_45231_1&content_id=_3245032_1

THE FOUNDATION

Substances in Unity

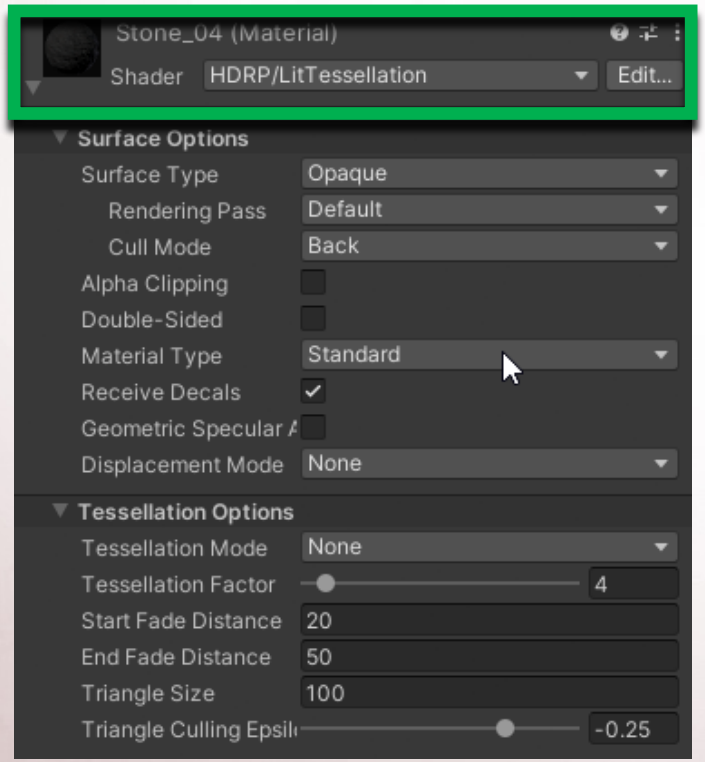
Setting up Unity's HDRP to use with Substances

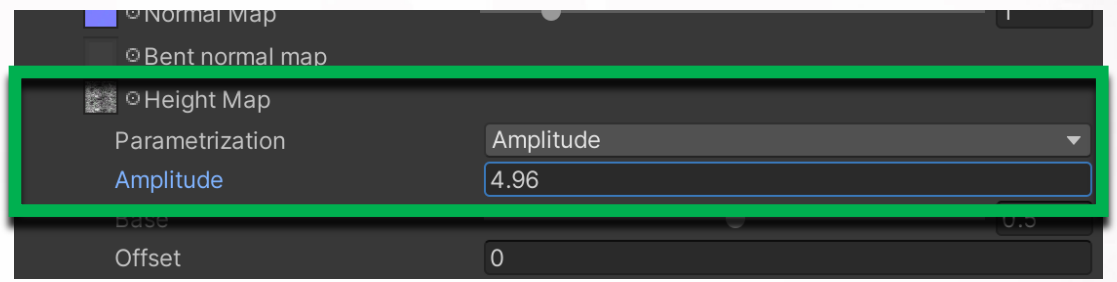
Publish your **.sbs** file (native substance graph file) as an **.sbsar** file inside of Designer (**right-click the .sbs to the left**). Install the Substance 3D in Unity plugin, then put your substances in your Unity projects, drag&drop those .**sbsar** files onto 3D assets (custom or primitives) and tweak the settings.

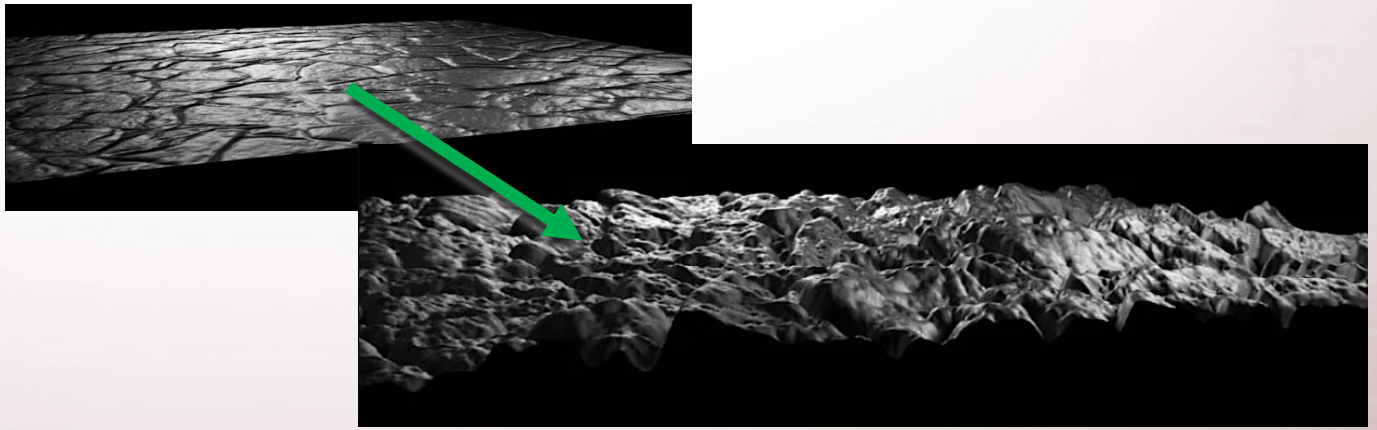
THE FOUNDATION

Substances in Unity

Tessellation (3D geometry) via **LitTessalation** shader (HDRP only, done through the Height map)







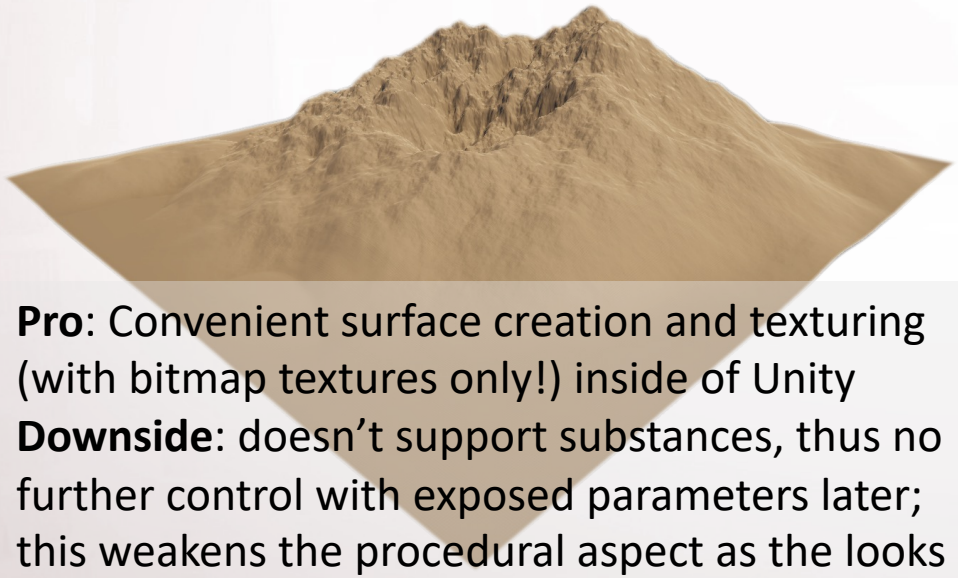
DRESSING THE STAGE

Terrain as "terrain" or as mesh?

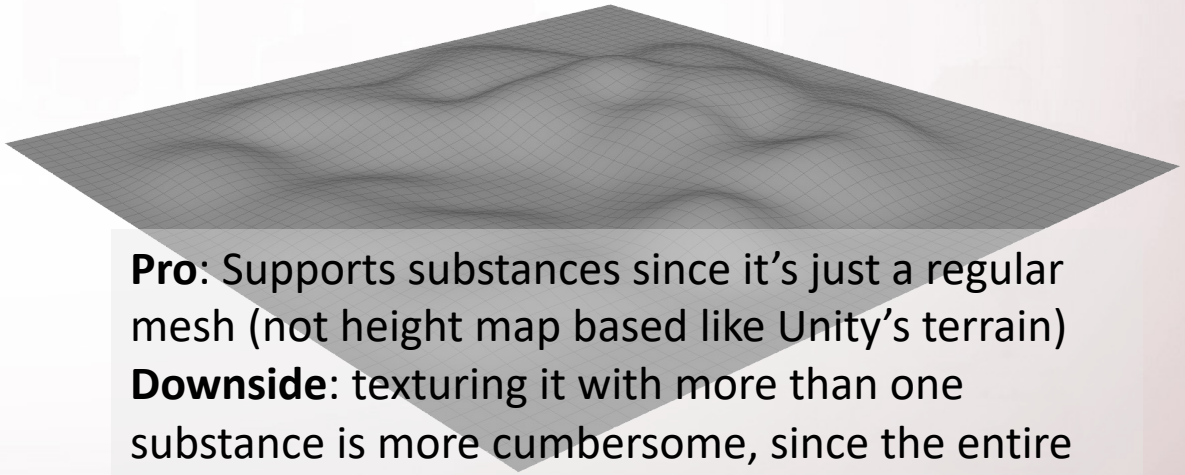
Unity Terrain

"VS"

Custom 3D Mesh



Pro: Convenient surface creation and texturing (with bitmap textures only!) inside of Unity
Downside: doesn't support substances, thus no further control with exposed parameters later; this weakens the procedural aspect as the looks of the terrain will be static and 'set in stone'

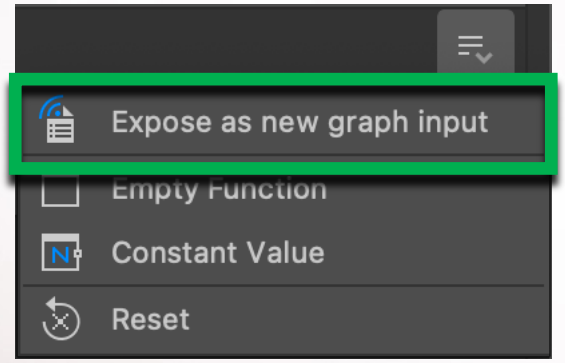


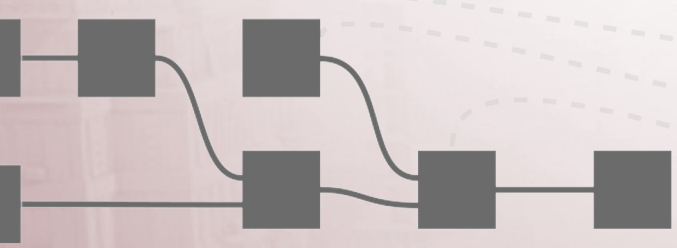
Pro: Supports substances since it's just a regular mesh (not height map based like Unity's terrain)
Downside: texturing it with more than one substance is more cumbersome, since the entire mesh will be tiling that one substance you applied to it (needs either extraction of polygons, thus separate meshes to create e.g. roads with another substance applied to or perhaps a Unity terrain or mesh underneath, penetrating the custom 3D terrain)

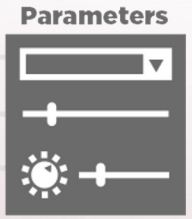
EXPOSING PARAMETERS

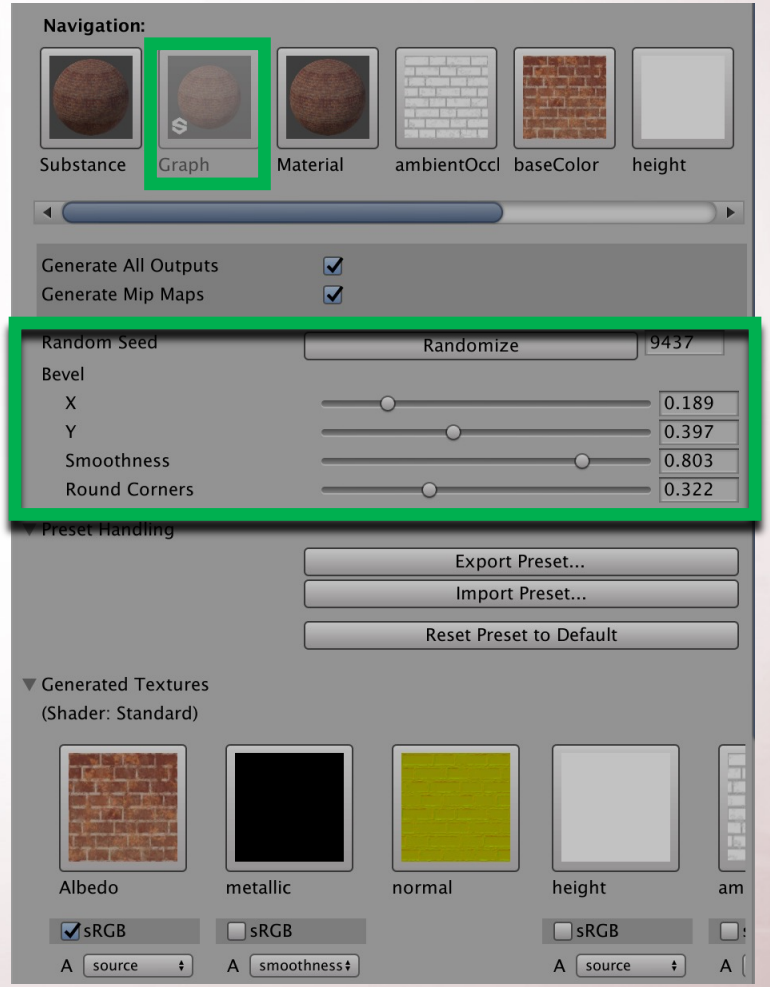
More about this

in the upcoming labs









Resources

REFERENCE LIST

- Fenech, B., Carter, J.T. (2017). **Approaches to Modular Construction for Real-time Game Environments** [PDF]. Canberra: University of Canberra. Retrieved from https://static1.squarespace.com/static/574e757260b5e93b27fb18ac/t/5b4804012b6a286d041b6fc9/1531446278799/BraidenFenech_ApproachesToModularConstructionForReal-timeGameEnvironments.pdf
- ***Using the Substance plugin in Unity*** [Video file]. (2018, April 26). Retrieved from https://youtu.be/Dy177PrxRfs
- ***Substance in Unity: HDRP Workflow*** [Video file]. (2020, June 2). Retrieved from https://youtu.be/k563_h0EdCI
- **Sculpting a basic landscape: Maya 2016 Essential Training** [Video file]. (2016, April 1). Retrieved from https://youtu.be/QBCnEsdr-sY
- ***How to make Terrain in Unity!*** [Video file]. (2019, June 23). Retrieved from https://youtu.be/MWQv2Bagwgk
- **Substance Designer - Exposing a Parameter**. (n.d.). Retrieved from https://docs.substance3d.com/sddoc/exposing-a-parameter-102400062.htm
- ***Convert your Old Unity Project from Built-In Render to High Definition Render pipeline*** [Video file]. (2018, Dec 2). Retrieved from https://youtu.be/orzuSWtaOVM

Time left?

SUBSTANCE CREATION TIME

What do we want to
create / showcase / explain / uncover / expose next lab?

Vote for the next substance in the next lab.
Poll starts in a minute...

Winner is: **Roof tile substance**

See you in the next lab!