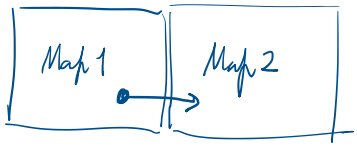
Section 2

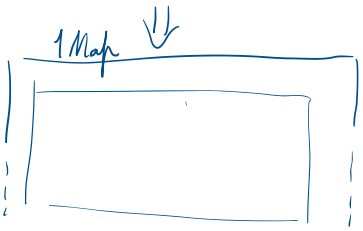
29 de junho de 2023 16:39

UE5 Bridge : A LOT OF FREE ASSETS

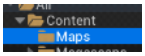
Large Worlds :



LOAD AND UNLOAD

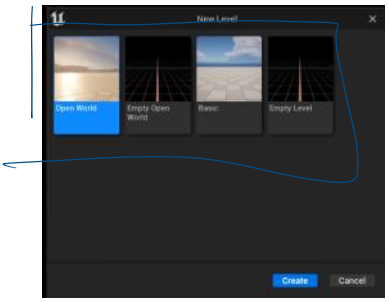


→ UE5 World partitioning



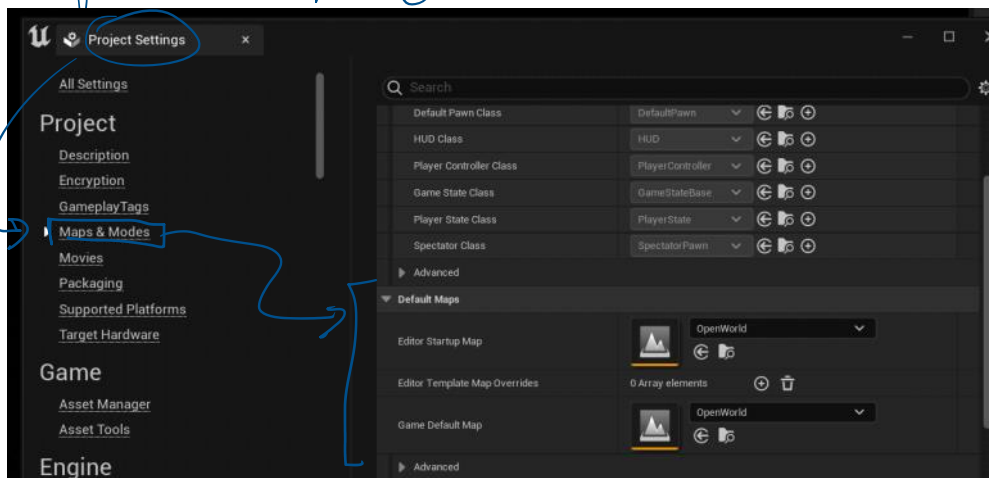
New Folder

↳ New level creation :

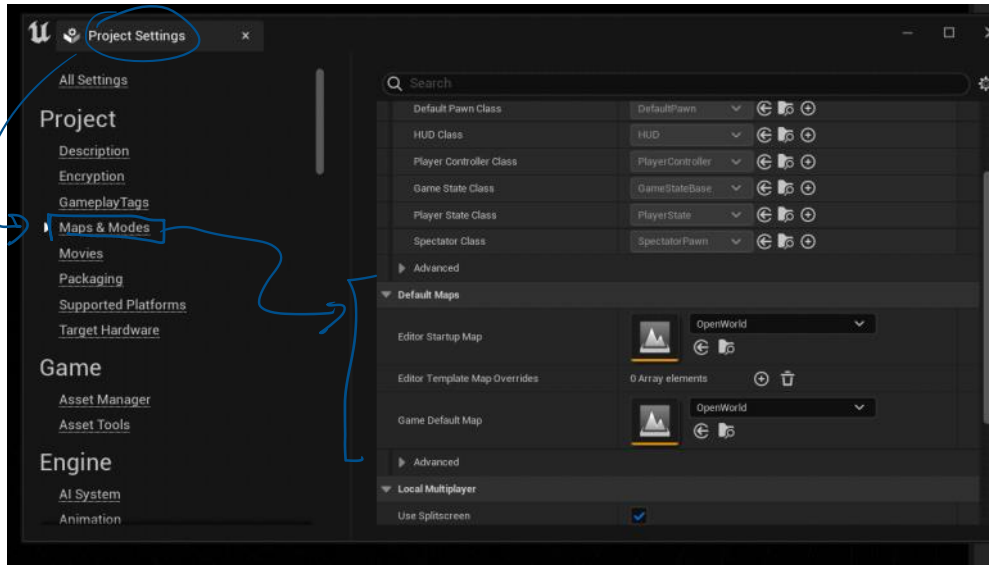


- Lighting and Atmosphere
  - Sky Atmosphere → Earth-like atmosphere Scatters light like a real one Up to 2 atm. lights
  - Directional Light → infinitely far away ⇒ shadows are PARALLEL (simulates the sun/moon)
  - Sky Light
  - Exponential Height Fog
  - Volumetric Clouds

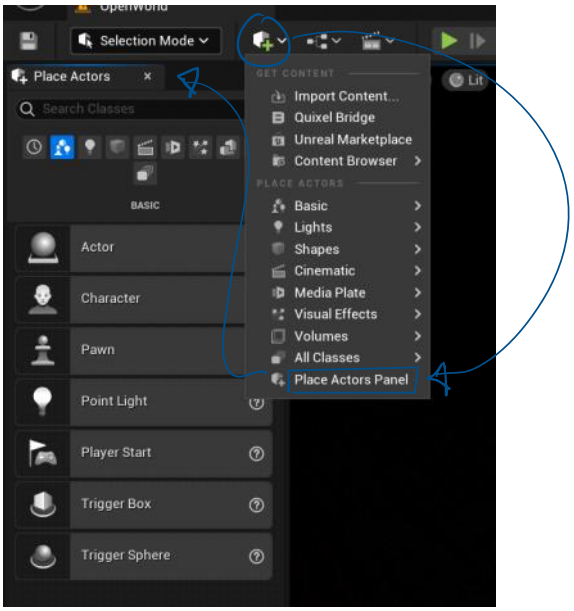
Default map when opening Editor :

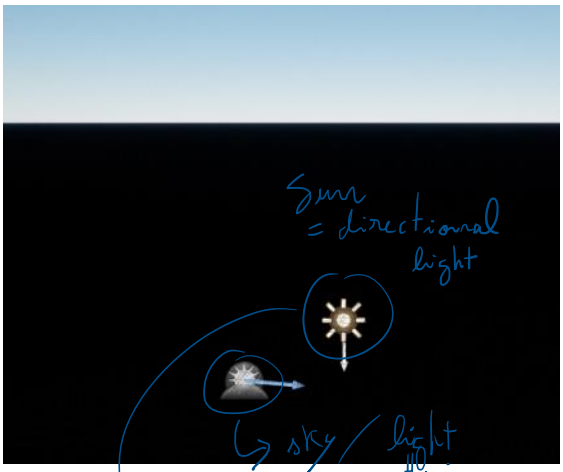


U



Place Actors Panel:





CTRL + L + mouse movement
⇒ move the sun (hold ctrl)

CTRL + Shift + L + mouse movement
⇒ moves the SECOND sun

[the rotation can also be changed manually]

Mobility:

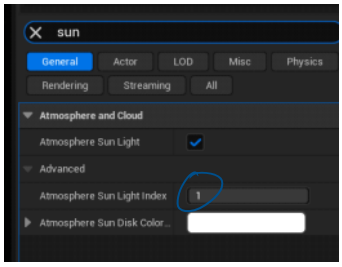
- Static
  - Lighting cannot be changed in-game

Mobility:

- **Static**
  - Lighting cannot be changed in-game
  - Fastest
  - Allows baked lighting
- **Stationary**
  - Color and Intensity can change in-game
  - Allows partially-baked lighting
- **Movable**
  - Can be moved and changed in-game
  - Dynamic shadows

} shadows for static objects

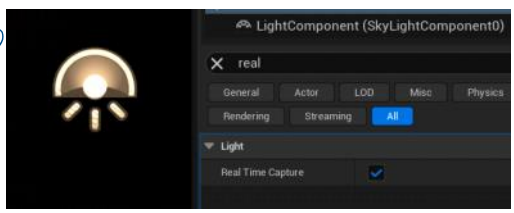
Change index of the second sun:



Sky Light

- Captures distant parts of the level
- Applies to the scene as light (reflections)
- Global illumination
- "Captures" only in certain conditions:
  - For static lights, updates when building lighting
  - For stationary and movable lights, updates once on load and on "Capture"
  - Constantly when Real-Time Capture is enabled

sky light sunset ≠ sky light mid day



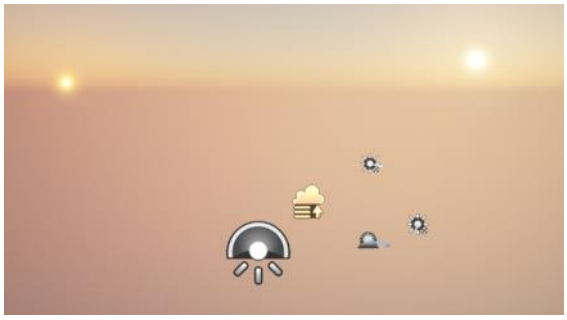
Fog and Clouds

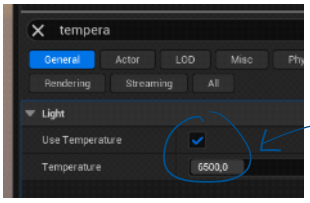
Exponential Height Fog

- Simulates Fog
- Gets thicker the lower you go
- Two colors
  - Hemisphere of a planet facing the sun
  - The other hemisphere

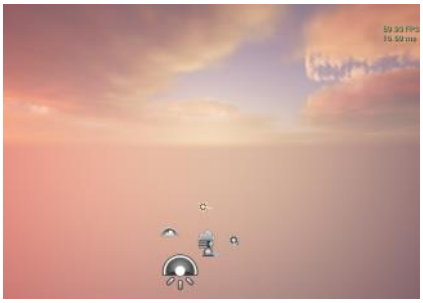
Volumetric Clouds

- Dynamic Clouds
- 3-dimensional
- Material driven
- Light-scattering

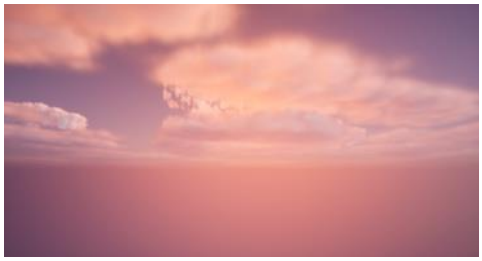




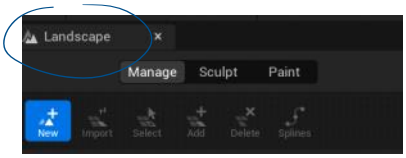
Temperature of the sun affects its color



Low temp sun (orange/red)
+
High temp sun (blue)
⇒ a purple color

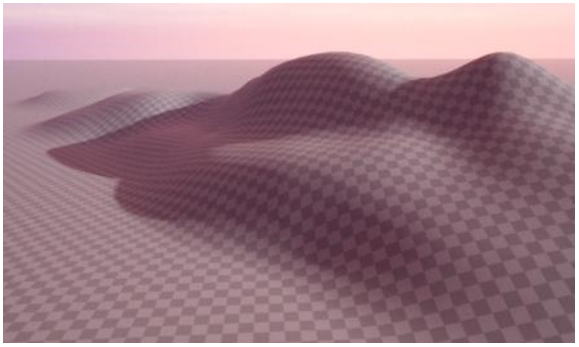


LANDSCAPE MODE



sculpt mode :

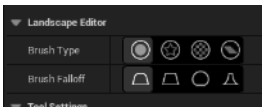
sculpt mode :

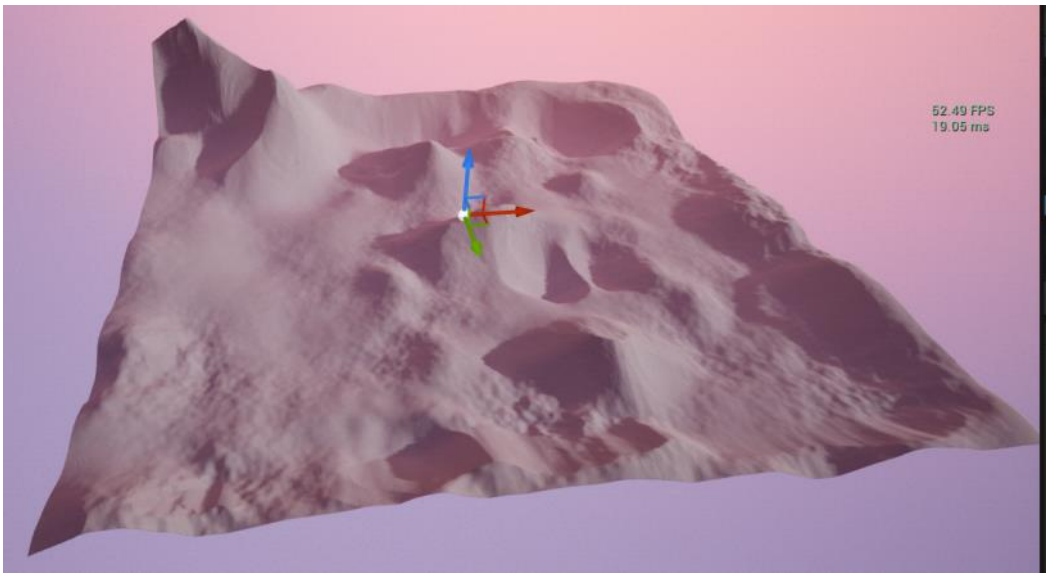


(Shift + 1 to go back to selection mode)

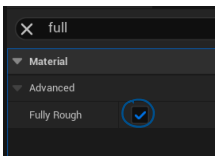
(Shift + 2 → landscape mode)

Brush Types:

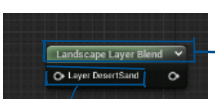




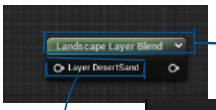
→ New Material starts with "M" (Material) → good/common practice



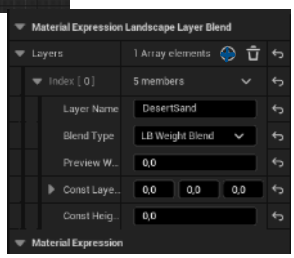
→ Not reflexive



→ Multiple Layers for the landscape

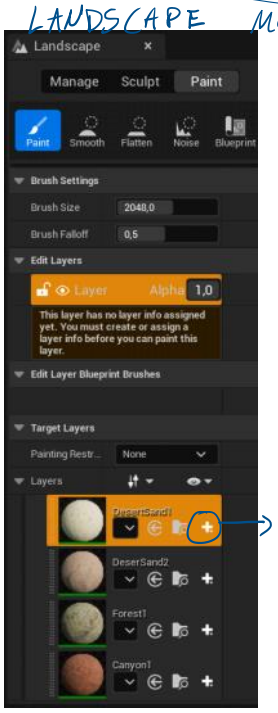


Multiple Layers for the landscape

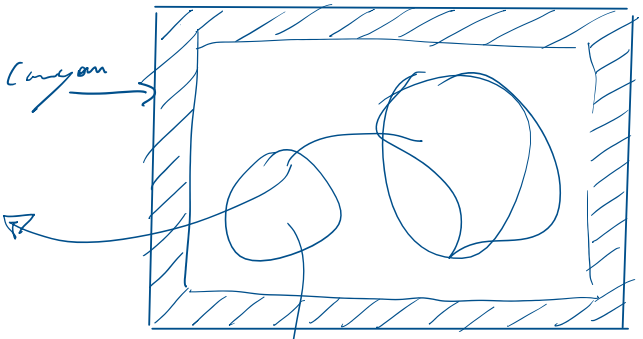


Correct colors and normals

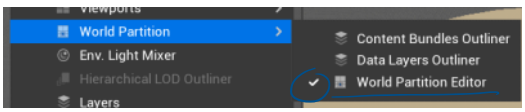
LANDSCAPE MODE

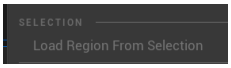


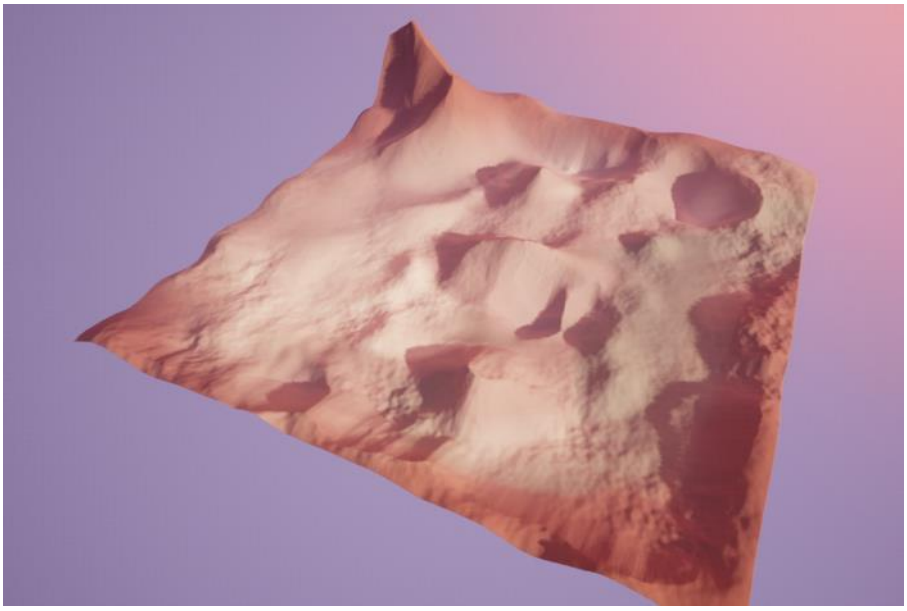
Create Layer Info



Windows

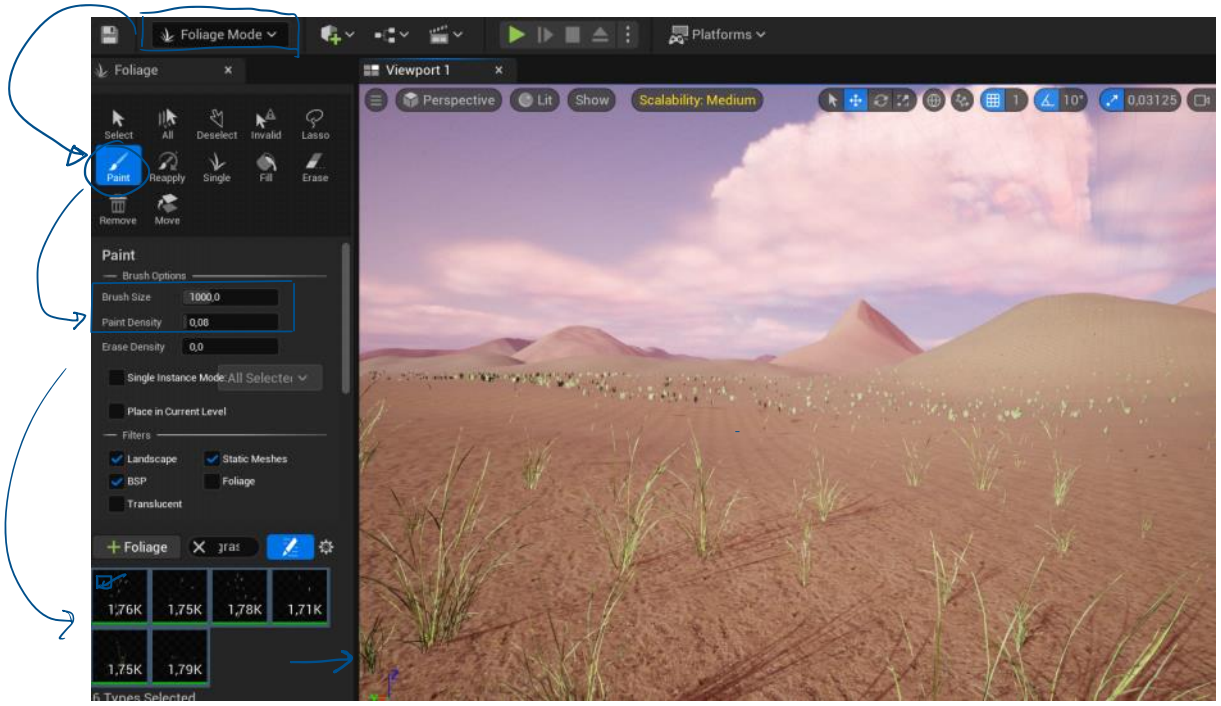




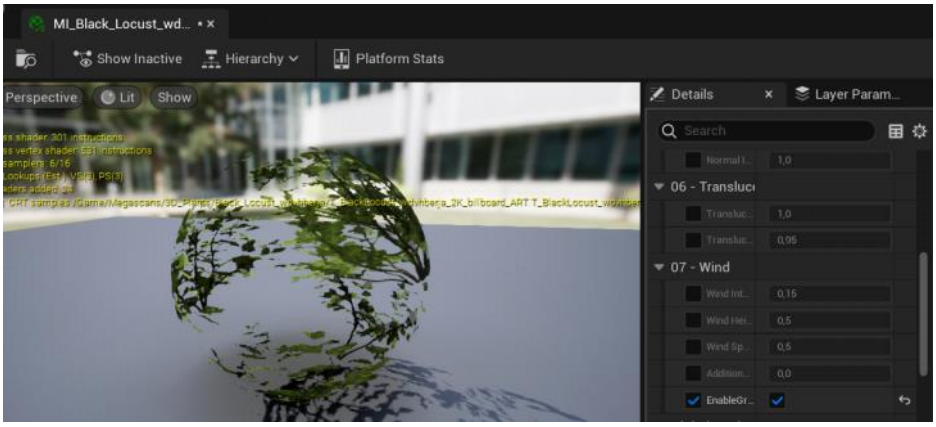


- FOLIAGE (my Quixel bridge assets)
  - Thatching grass
  - Wheat grass

Foliage instances => share most data => efficiency

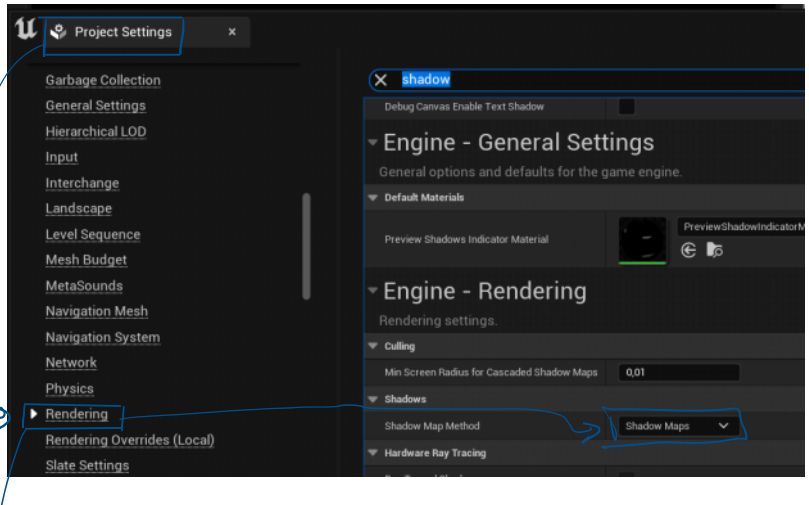


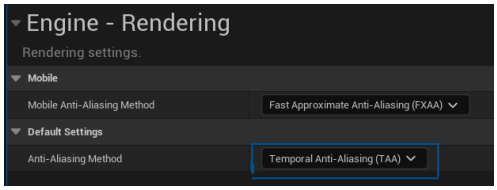
To enable WIND:



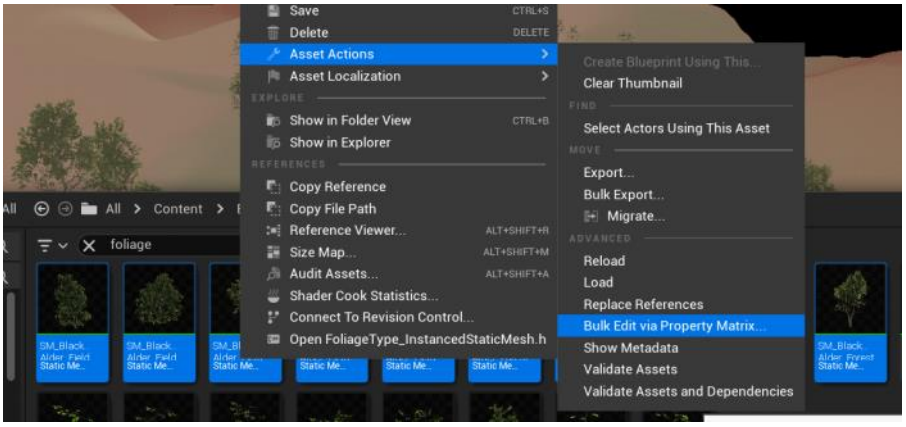
→ uncheck for big trees

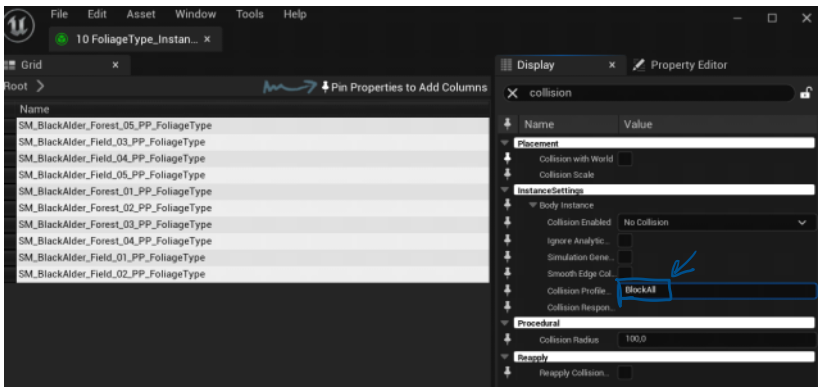
Increase frame rate :





Activate colision (trees) :





GROUP
- TRANSFORMS → TYPE
  - XYZ
  - ANGLE
- MODELS

USAR CLASSES :)

USAR ENUM :)

TREE parseGroups (group) {

ADD
- TRANSFORMAÇÕES
- MODELS
- GROUPS
  - CHAMAR parseGroups em cada um
    - ⇒ RECURSIVO

```
for(TiXmlElement* chGroup = group->FirstChild
    Tree child = parseGroups(chGroup);
    // Tratar de appends do filho para a árvo
    addTreeChild(res, child);
}
return res;
```

ENGINE

void drawGroups (Tree groups):

- APLICAR TRANSFORMAÇÕES
- CHAMAR DRAWFIGURAS (models)
- CHAMAR PARA OS NODOS FILHO

```
List filhos = getChildren(groups);
for(unsigned long i = 0; i < getListLength(filhos); i++){
    Tree next = (Tree)getListElemAt(filhos, i);
    drawGroups(next);
}
        glPopMatrix(); // retorna ao respetivo estado anterior dos eixos.
}
```