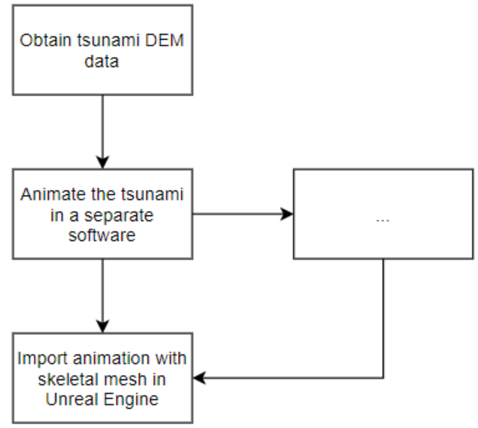
## 1. Animating the tsunami using the heightmaps

**Idea**: animate a plane in a different software, then import the animation in Unreal Engine




**Pros**:

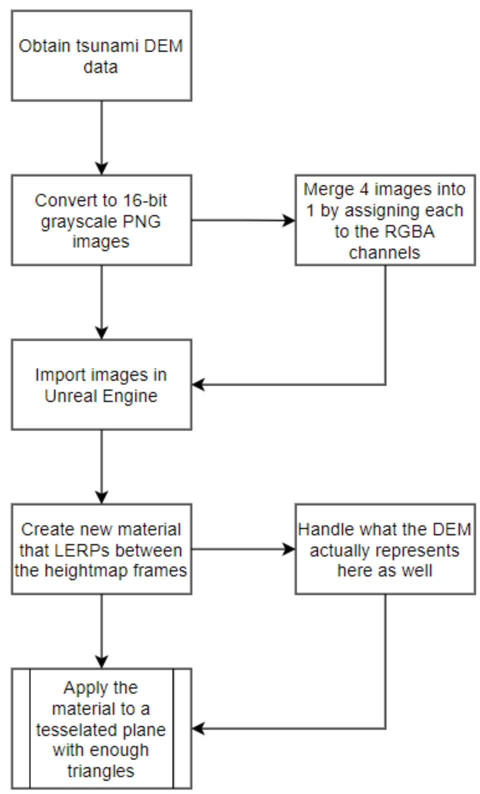
- Adding physics is simpler?
- Less processing power is used to animate the tsunami than procedurally animating it
- Less work done in Unreal Engine

**Cons**:

- More software needed than UE
- More difficult to communicate between mesh and landscape underneath?
- Heightmap data either needs to be small or low resolution (UE has limits on how big images can be)
- Needs to be manually aligned with landscape (geographic location of the tsunami DEMs do not match the landscape DEM)

## 2. Interpolating between tsunami heightmaps (what I showed in our last meeting)

**Idea**: animate the tsunami using UE materials

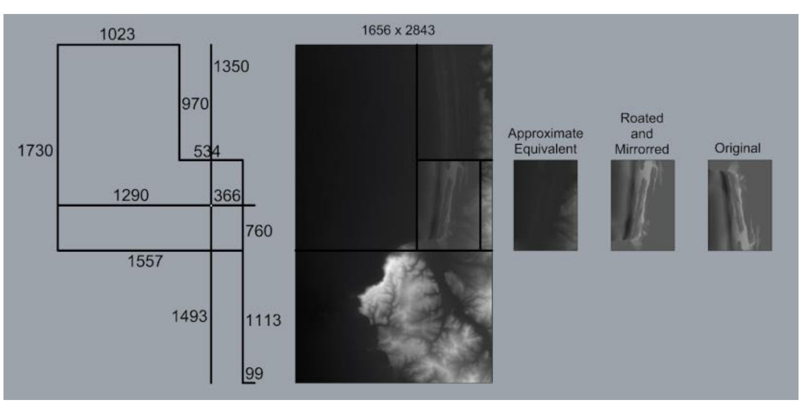



**Pros**:

- Easy and simple workflow
- Easier to communicate between landscape and mesh (using render targets?)

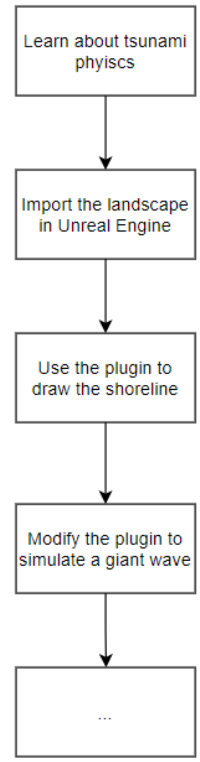
**Cons**:

- Need to use other software to convert DEMs into images
- Tedious to implement (simplified using material functions)
- Difficult to handle normals? (resulting material looks flat, difficult to see details)
- Requires more processing power (30+ textures for a single material, past the ideal 16 texture limit)
- Heightmap data either needs to be small or low resolution (UE has limits on how big images can be)
- Needs to be manually aligned with landscape (geographic location of the tsunami DEMs do not match the landscape DEM)



## 3. Using the Water plugin from UE 4.26

**Idea**: use the Water plugin to simulate a tsunami wave




**Pros**:

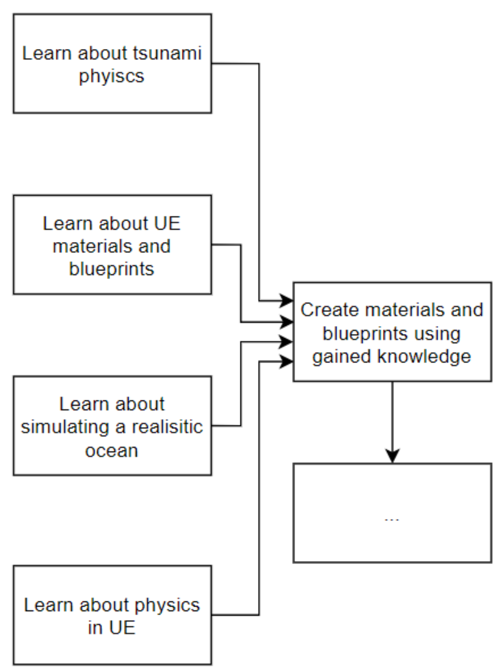
- Basic shallow waves are implemented (waves die down when the reach the shore)
- Lots of settings can be tweaked to affect ocean shape
- Convenient for quickly simulating water bodies
- Physics with water velocity and buoyancy can be used (using WPOs)
- Very easy to communicate between landscape and mesh
- Semi-realistic ocean already implemented
- Easy to identify the shoreline

**Cons**:

- Need to use other software to convert DEMs into images
- Complex shallow wave physics are not implemented (required for tsunami simulation)
- Beach environment needs to be manually created (waves that automatically go to the shore the further away they get from the deep ocean)
- Requires advanced knowledge of Unreal Engine materials and blueprints (especially with how the plugin works)
- The plugin's materials and blueprints will need to be manually adjusted, which might affect other parts of the plugin
- Water is not able to go beyond the shore (this has to be manually adjusted – is this possible?)
- VERY buggy
- Not meant for large landscapes
- Works best with uniform landscape scales (ie. X,Y,Z all have the same scale factor)

## 4. Using custom materials/blueprints

**Idea**: simulate the tsunami by manually creating the materials and blueprints




**Pros**:

- Full control over what is being simulated
- Full control over what physics is implemented
- Can be used to accurately simulate historic and future tsunamis with only a few parameters
- Very scalable with regards to simulating different kinds of tsunamis based on a terrain
- No need to pre-animate heightmaps

**Cons**:

- Rather than using heightmaps, data on tsunami wavelength and ocean depth would be needed
- Need to learn how to use UE materials
- Need to learn about tsunami physics
- Need to learn about simulating a realistic ocean
- Need to learn how to communicate between landscape and mesh (or how to identify the shoreline and heightmap data)
- Need to learn how to apply physics (buoyancy, destroying objects, etc.)
- Not as scalable with anything beyond ocean, lake, and tsunami simulation

## Things I need to learn and do:

1. Learn how to use Render Targets
2. Learn how to read ocean depth in material (something to do with blueprints and render targets)
3. Learn how to get distance from water position to shore (to figure out how wave height changes as it gets closer)
4. Affect a wave's height, wavelength, and steepness based on depth (shallow water waves)
5. Figure out how to send water body information (position, velocity, depth, etc.) out of a material
6. Add realism to water material
    a. many waves at once
    b. wave foam based on wave height
        i. no foam on calm sea
    c. caustics and light refraction
    d. figure out how deep water waves transition to shallow water waves (and how they generally face the same direction – wave refraction)
    e. fake transparency (does this work with light refractions?)
    f. two sided triangles
    g. water color based on depth/height
    h. toggle between crest-first or trough-first waves (both using sine)
    i. procedural particles for breaking waves

## What I know so far about Wave Physics

Wavelength: L
Amplitude: A
Speed: c
Wave Height: H
Wave Base: B = L/2
Water Depth: D
Gravity: g
---------------

if D < L/20 (or 20*D < L)

**SHALLOW WAVES**
c = sqrt(g*D)
H:
Airy linear wave theory: H = A*D / B
H = K_s * D
K_s = A / B
More Common??? H = A * (B / D) ^ .25
H_shallow / H_deep = (D_deep / D_shallow) ^ .25
H / H_start = (D_start / D) ^ .25
H / A = (B / D) ^ .25

else if L/2 > D > L/20 (or 2*D < L && 20*D > L)
**INTERMEDIATE/TRANSITIONAL WAVES**
c = sqrt(g*L/2pi * tanh(2*pi*D/L))
H:
Airy linear wave theory: H = A*D / B
H = K_s * D
K_s = A / B
More Common??? H = A * (B / D) ^ .25
H_shallow / H_deep = (D_deep / D_shallow) ^ .25
H / H_start = (D_start / D) ^ .25
H / A = (B / D) ^ .25
else if D > 0

**DEEP WAVES**

c = sqrt(g*L/2pi)

H = A

"About half of the waves in the open sea are less than 2 m high, and only 10-15% exceed 6 m."

"Under strong wind conditions, the ocean surface becomes a chaotic mixture of choppy, whitecapped wind-generated waves."

"A fully developed sea often occurs under stormy conditions, where high winds create a chaotic, random pattern of waves and whitecaps of varying sizes."

else

**NO WATER**

---------------

if H/L > 1/7 (or 7H > L)

BREAK WAVE

decrease H (using gravity) if XY_OFFSET (gerstner waves also offset X and Y, not just Z) is past a certain threshold

---------------

TSUNAMI:

- shallow water wave, but with custom wave base and wavelength
- wave base must be greater than water depth
    - equal to water depth? multiply by 2 to get wavelength?
- original height (wave amplitude?) is determined by the amount by which the sea-floor got displaced
- shallow water height determined by following equation:

    H_shallow / H_deep = (D_deep / D_shallow) ^ .25

    H_pos / H_start = (D_start / D_pos) ^ .25

    H_pos = H_start * (D_start / D_pos) ^ .25
- water height over land is determined by following equation:

    ???

---------------

CAPILLARY WAVES:

- use a normal map rather than vertex displacement

WAVE REFRACTION:

- adjust wave direction based on how close it is to the shore (clamping after it is perpendicular) and how perpendicular the wave direction is with the shore