PROJECT PLAN (very casual)

- Seasons
  - Spring
    - Smaller, budding leaves
    - Maybe white/pink looking leaves too to symbolize flowers?
  - Summer
    - Full green leaves
  - Autumn
    - Colored leaves (orange, red, yellow)
    - Winter
    - No leaves
    - Frosty frame?
- Level of Recursion in Branches
- Different rules/axioms type of tree
- Basic L-System (library where you can declare axiom, rule, and query for a string depending on recursion)
- Stochastic rules for L-System (probability of selecting rule)
- Potentially…
  - branch radial decay
  - level of randomness of change in angle at each successive branch

Technical for Tree.

- L-System - Lucy
  - Levels of Recursion
  - Stochasticity
- L-System -> Graphic - Griffin + Katherine
  - Positioning branches
    - Having a set angle, but being able to change it
    - Stochasticity in angles
  - Positioning leaves
    - number of leaves per branch
- Graphic
  - Parameterizing Leaf (size, color)
  - Parametrizing Branch (radial decay, radius size, length, color)

Resources for L Systems

- Not sure if we are going to generate our trees via creating a mesh (L System strings → triangle vertices) or primitives (strings -> primitives), for the latter, the geometry shader may be useful
  - Either way, would be good to support branch tapering

- This is long but worth skimming, but I like how they mentioned that a good primitive to use would be a cylinder or cone, with a half-sphere at the end to create a closed mesh? I think the ends of the branches will also look prettier this way. Anyways, it may be interesting to be able to choose between different primitive bases if we choose to go that route.
- A brief lecture on l-systems and trees
  - We will probably need a LSystem class, that given an alphabet, rules, and axiom, generates some string. Options for stochastic + deterministic doesn't seem too difficult.
  - Given the output of the LSystem class, transform the strings into the 3D model. Can add stochasticity here as well (branch radial and length tapering, variations in angle, # of children)
- Procedural Plant Generation with L Systems
- http://algorithmicbotany.org/papers/#abop
- LSystems and their outputs (2d) ***could be helpful to build our parser***
- https://www.researchgate.net/publication/216337877_Intelligent_Tree_Modeling_Based_on_L-system?fbclid=IwAR1pv_5PAZ9hHtLgYGWhc83Z2J8VRgrFBahXPAls_2jv4cAznBiwiXJlmVM

Bump mapping Resources

- https://learnopengl.com/Advanced-Lighting/Normal-Mapping
- http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-13-normal-mapping/
  - Basically, in addition to storing the colors of a texture map, we will now be computing a normal, which will be based on the image color at that location.
- https://www.youtube.com/watch?v=hOLLh80hDmw&ab_channel=BrianWill

Seasons ideas

- Toggles?
- Snow particles/frosted glass for winter?
- Spring/Autumn: Change color of leaves

ORIGINAL PROJECT PROPOSAL

CS Logins: ksang1, lqu2, gkupsaw
Github: https://github.com/artset/cs123-final
We have two proposals

A tree through different seasons:

Features

- The user can toggle the change of the tree’s appearance through the seasons

    - For instance, green leaves budding in spring, larger green leaves in summer, orange/brown leaves in autumn, and snow on branches in winter
    - Lighting will change throughout the seasons (more sun in the warmer seasons, vs grey light in the winter)
- Leaves growing and falling through different seasons
- User can potentially toggle other aspects about the tree (Thickness of branches, randomness)

Technical Components

- L Systems for tree structure
- Bump mapping / displacement mapping for the trunk texture
- Environmental mapping

Snow simulation (Snow globe / snow on a terrain)

Features

- Different lighting (day/night)
- Wind simulation (and snow collecting in different areas over time bc of wind)
- Parameters of snow (hardness, weight, chunkiness, amount)
- Extra: if the snow is in a “box” maybe, being able to rotate/”shake” the box to unsettle the snow.
- Potentially doing snow on a terrain and generating the terrain.

Technical components

- Collision detection
- Deferred Lighting
- Environmental mapping
- Terrain generation