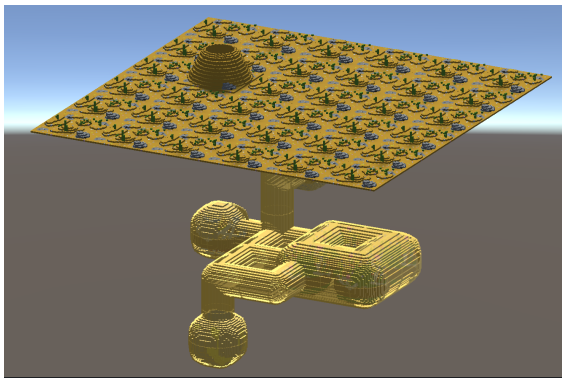
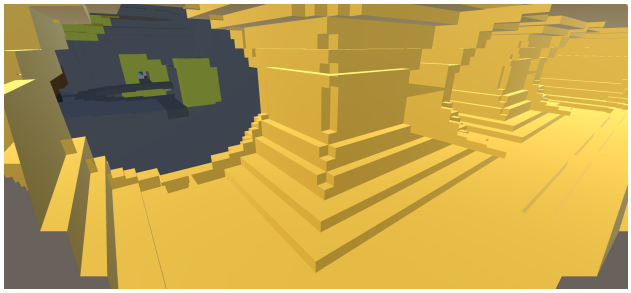
CSCI 5529 Graduate Project Proposal
Brian Glassman

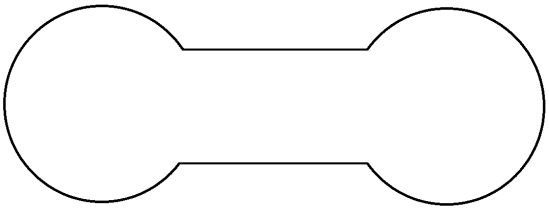
Summary: I will be upgrading the visualization elements for a game I am making about managing an ant colony. This will be a replacement for the visuals currently in use, which are voxel-based, fairly simplistic, untextured, and use very basic lighting. The image on the left below shows the colony from the outside, while the image on the right shows an interior, first-person view. The player will be able to access both views, as well as move around inside them.

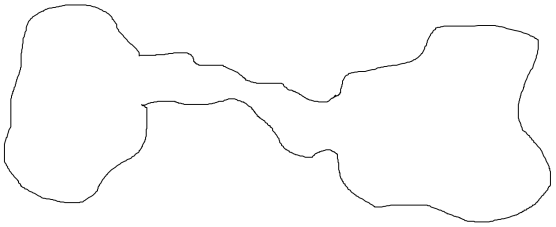




The upgrade will use meshes to give more visual detail and avoid the squared off look of voxels, will use more complex geometry for the tunnels and chambers, and will include realistic textures and lighting. The upgrade will also include a cutaway effect that allows seeing into the tunnels and chambers from the outside, either using culled faces or a shader depending on time constraints.

In the current version the structures use a small number of pre-generated models with very rigid connections between them, while the upgraded version will use a procedural mesh that includes random noise. This will mean that instead of straight-line tunnels with perfectly straight walls, the tunnels will meander, change size and shape, and have uneven walls. A 2D view of the current version is shown on the left below, with a mockup of the upgraded version of the same layout on the right. The largest effort of this project will likely be creating this mesh and ensuring that the textures and normals are handled correctly.





The existing implementation is in Unity, but for this project I will be starting from scratch using C++ and OpenGL.

Capabilities: The final product will be an interactive construction sandbox, although for the purposes of this project there will be no gameplay elements implemented. The user will be able to place new tunnels or chambers in a 3D grid and have the appropriate connections to adjacent structures form automatically. The user will be able to orbit the colony in the exterior view, with cutaways giving a view inside the chambers and tunnels. An alternate first person view will also be available, allowing the user to explore the colony from the inside. For visual variety as well as future gameplay purposes, the colony will be populated by other ants and artificial structures such as mining or farming chambers.

Implementation: The chambers and tunnels will both be implemented as C++ classes, using the class to handle connections with adjacent structures and storing the computed mesh information. To aid development and review there will be options available to visualize the underlying mesh both with and without this noise effect. There will also be an option to load a pre-designed colony, rather than creating one from scratch.

Definite Goals and Schedule:
Completed by initial progress report

- Predefined layout
- Meshes for tunnels and empty chambers, with adjacent structures connecting to each other
- Placeholder textures and lighting, to be further improved as the mesh is finalized

Completed by Thanksgiving review

- Random noise perturbations added
- Buildings and other ants to populate the scene
- Exterior and interior views fully implemented, with user controls to move around in these views

Completed by final submission

- Interactivity, allowing the user to place new structures or delete existing ones
- UI menu to select which structure to place (options will be: tunnel, farm, mine, delete)
- 3D grid visualization and a “ghost” of the structure to be placed, to let the user visualize the placement before confirming

Stretch Goals
Listed in order of increasing (presumed) difficulty:

- Texture variations, like different types of ores in the walls
- Merging adjacent chambers, rather than connecting them with short tunnels
    - The current implementation has this capability, shown in the four connected chambers in the exterior view, but it will need significant rework with the new mesh-based system
- Add shadowing, including shadows from when multiple light sources overlap
- Using a custom shader for the cutaways in the exterior view to create a smooth cut, rather than the jarring visuals produced by face culling
- Add animations to the ants and buildings populating the colony
- Add collision detection in the interior view, so that the user cannot pass through the walls