The Question: If a player only has one color and the ability to create, will that player live in harmony with another in order to make something beautiful, or destroy to show dominance?

Our Goals:

* Simple creation tools
* Grass, Sheep, Wolves triangle
* Bright colors
* Expansive Biome
* Height variation

Description:

We want to create something unique.

It will be a world where creativity is possible, but where others can ruin your fun at any moment.

Sensation of setting dominoes in a crowded room.

Group:

* Member Organization/Modeling
  + David Kahrs
* Main Mechanics / AI
  + Nick Sheets
* Debugging/Networking Solutions
  + Ian Kane

We plan to share the work on all fronts, this is just a start.

Plans:

* Expansive workable area
* 4 players using basic colors
* Many tiles under control for each player
* Overhead view, such as RTS games
* Defined tiles
* Spawning Items
* Creature AI
  + Wolf vs Sheep vs Grass
* Clients
  + Detect player input
  + Send requests to instantiate their tools
  + Display visuals
* Server
  + Maintain current state of the world
  + Perform AI algorithms
  + Relay information between players
    - Chat
* Dynamic Terrain Modification
  + Is it possible to modify terrain meshes via dynamic code?
  + What have others done and what can we do?
* Workable product
* Successful dynamic spawning of objects
* Object AI
* Networking Resolved
  + Such as players disconnecting

Luxuries:

* Animations/Modeling
* Balancing

Development Changes:

* Ditched chat: We thought that in order to make the game more natural, there shouldn’t be talking between players, and that alliances and enemies will be formed just through gameplay. The gameplay does most of that talking for the player.
* Tile Structure: The only thing bound to the grid is the grass blocks and the height variation. Animals are no longer tile based, but work on top of tiles.

Development issues person by Person:

Ian kane:

Sheep Colors: changing the sheep colors for every player on every sheep spawn was really hard to figure out. First of all, the material was on the Polycontainer, not the actual model, so we had to get the child of the model. After that, we had to find a way to apply the color material to every sheep upon it’s creation.

Nick Sheets:

The main complications I faced during this project were learning terrain coordinate systems, as well as trying to interact with specific models that did not have network views, but being able to pass GameObject references over the network. Additionally I faced issues of terrain data (such as trying to modify details which is a nightmare).

David Kahrs:

There were several issues with the GUI system in regards to displayed names and menu navigation. The names were being improperly stored and there was no call to update the names everywhere. Also, there were issues with resetting settings when a player connected/disconnected. At the color selection screen, one player would be able to connect, select a color, disconnect, and then connect and take up another color slot. This means that one player could prevent another playing from selecting another color.

Development Achievements:  
  
Ian Kane

Made all of the colors work. Made the Network IDList, Set Up all of the infrastructure and converted class code for our own use. Set up the Economy, tracks how many of each unit is active and tells the server.

Nick Sheets

I was responsible for the AI of grass, wolves, and sheep, the terrain modification code, as well as building the server control scheme so as to broadcast AI position and behavior across the network.

David Kahrs

I created the base models for the grass/sheep/wolf and I worked with the GUI system. The menus successfully work now. Also, I worked with the camera settings so the player has an easy time navigating the map.

Structure and technical setup:

All of the objects are on two 2d grids, one for terrain and one for everything else. All of the movement and positions are kept on the network, and metrics, such as the ability to place objects, is kept on the local machine. All AI Is handled by the server, and information is sent to clients.