Plans

* User Interface
* Hexmap Load/Save
  + Load hexmap on startup
  + Save hexmap on normal close
* Weather sampling
  + Use baseline HEX values and global values to sample weather for the hex
    - Save the generated weather for multiple samplings, reset each day
  + Smooth weather. Each HEX will have a set of values for weather. There will be defaults based off of the type of hex at creation
    - Average HUMIDITY between neighboring hexes
    - Water tiles create humidity, grasslands+forests are sinks?
* Hex General
  + Hex color should be determined based off of temperature, humidity, and altitude
    - biodiversity
    - temperature
    - rainfall
    - fertility
  + So a little function-doodle that calculates RGB
    - Altitude decreases saturation
    - Humidity scales green
    - Temperature scales red
  + Add notes to hexes
  + Add fun features to the hexes
  + Draw noteworthy features
  + New types of drawing tools
    - Selector
      * Single click selects one, opens up a context window
      * Click+drag to select large group
      * Cut/copy and paste?
    - Hand tool
      * click and drag moves
      * click does???
  + Different kinds of water
    - fresh
    - saline
    - potability ?
  + Have it discover regions of similar biomes
* Time tracking
  + Hexmap should keep track of time of day
  + Allow time skips
  + Keep track of seasons, moon phase, etc…
* Party
  + Keep track of party location
  + Allow travel.
    - Use local hex to determine travel time
    - Move time forward accordingly
* Procedural description
  + Use local data to provide a description of the landscape and weather
* World generation

Current ToDo:

* Add last couple hex properties
  + Coastal bool, map edge bool
* Remove superfluous hex/point methods
* Implement Save/Load
* Develop Procedural world generation
* Optimizations!!!

Generation Steps

* Seed “mountain tops.” These will be scattered uniformly throughout the region (SPARSE). Max altitude, or an altitude very close to max.
* At each step, generate neighbors for the existing tiles – generic tiles. Each generated one will sample an altitude from a Gaussian distribution slightly lower than the parent hex.
  + Save a bool so we can skip the hexes that are in the middle
* Repeat this process. When altitude reaches zero, generate ocean hexes until the whole region is filled up
* Now, we need to assign hex types.
  + Add lakes in places surrounded by mountain