The Heroes 3 Random Map Generator

Gus Smedstad
New World Computing

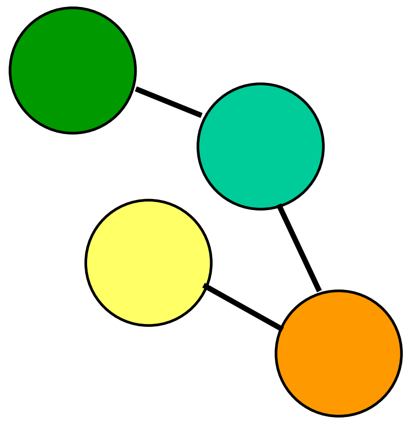
Goals

- Maps from 36x36x1 to 144x144x2
- Strategically balanced.
- Natural land shapes.
- Castles, Monsters, Treasures.
- Aesthetically pleasing obstacles.
- Rivers and roads

Example Map

Overview of generator

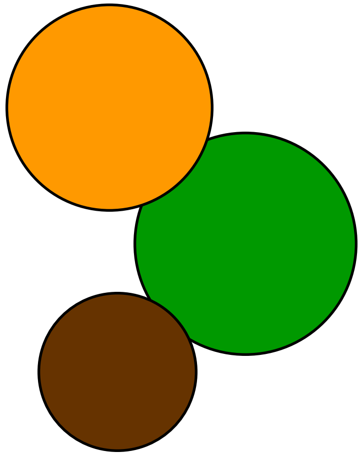
- Random Map Templates
- Zones
- Terrain placement
- Obstacle Mask
- Object Placement
- Creation of Connections
- Treasure selection
- Obstacle Creation



Random Map Templates

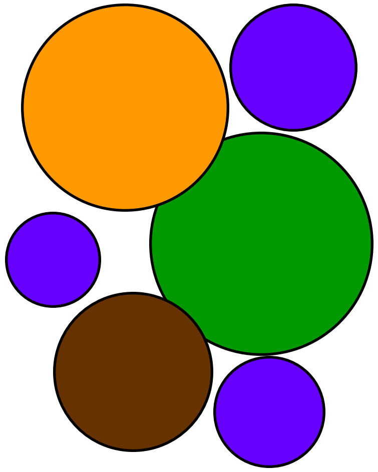
- Determine strategic shape of the map
- Zones
  - Type of zone: human start, computer start, treasure, junction
  - Zone size
  - Density, minimum number, type, and ownership of castles
  - Density, minimum number, and type of mines
  - Type of terrain
  - Density and value range for treasures
  - Strength of guarding monsters
- Connections between zones

Zone placement



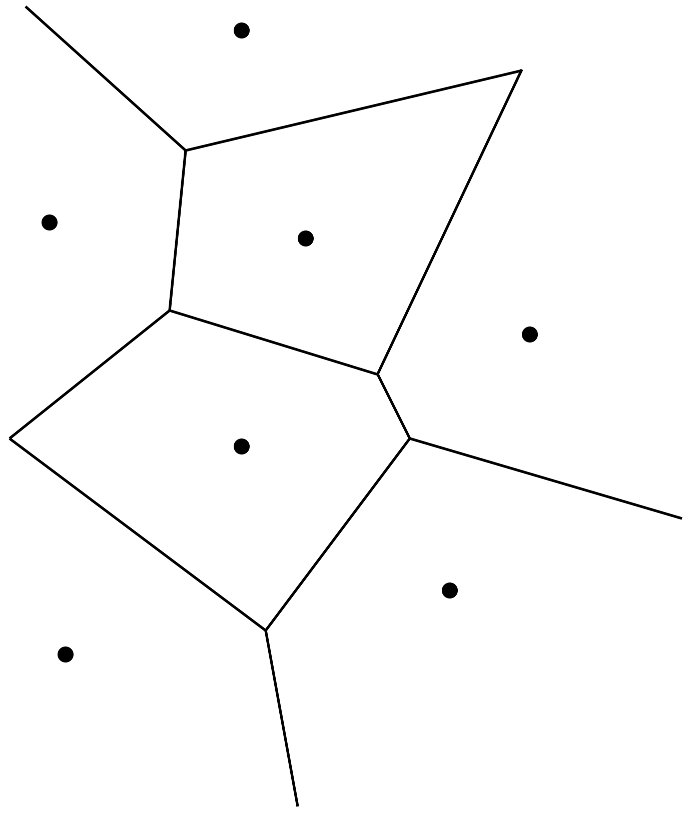
- Zones treated as circles at this stage
- Important concept at this stage is zone's center
- Placement done on arbitrary grid, translated and scaled later to actual map size
- First zone placed in center
- Later zones placed adjacent to existing zones
- Zones allowed a small amount of overlap

Zone Placement continued



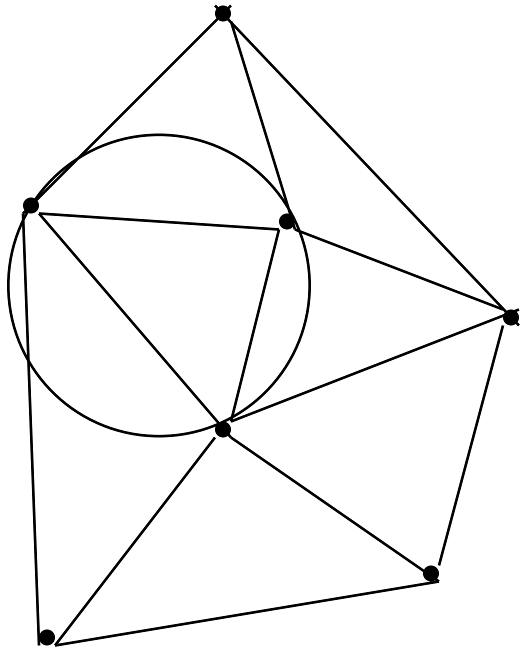
- Minimize overall width and height of map
- For two level maps, each level must have at least one zone
- Maximize number of adjacencies to connected zones
- Zone placement done 5 times, like “shaking a box”
- Extra water and rock zones placed after normal zones

Conversion of zones to terrain



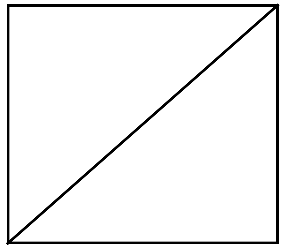
- Basic zone shape depends on adjacent zones, like soap bubbles
- Starting zone boundaries determined by Voronoi diagram of zone centers

Creation of Voronoi Diagrams

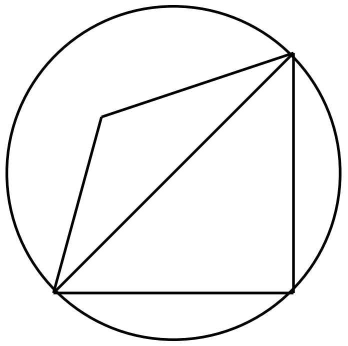


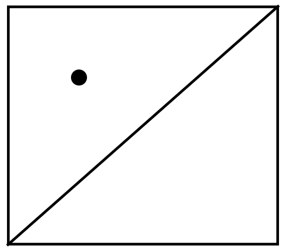
- Delaunay triangularization
  - Each triangle defines a circle
  - No circle contains any points
- Delaunay triangularization and Voronoi diagrams
  - Center of each circle is a vertex of the Voronoi diagram

Creation of Delaunay triangularization

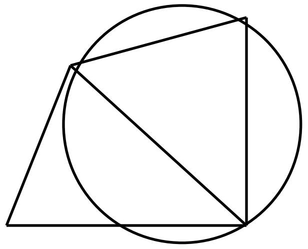


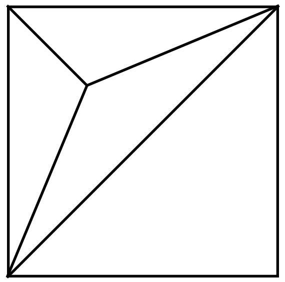
Start with 4 points defining a square outside the boundaries of the map





Insert each point into the diagram



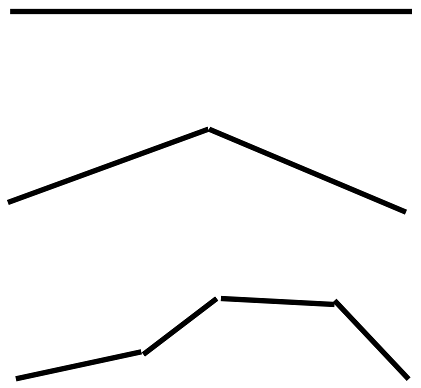


Create new triangles with new point as a vertex

Examine pairs of triangles and swap edges if necessary to create Delaunay triangles

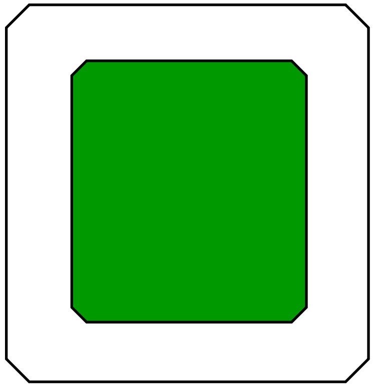
Creation of Zone edges

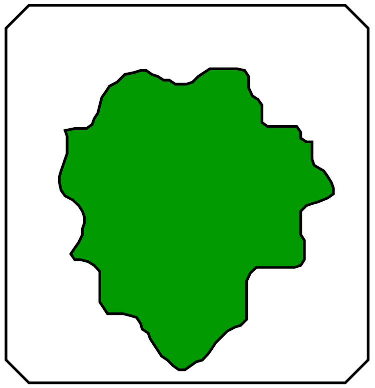
- Fractal randomization of zone boundaries
  - begin with line defining boundary
  - displace center by a random amount, up to 1/2 line length
  - repeat with two resulting lines



Differences on an Islands map

- Zone boundaries not randomized
- Internal boundary between land and sea within each zone
- Internal boundary displaced from actual boundary and randomized

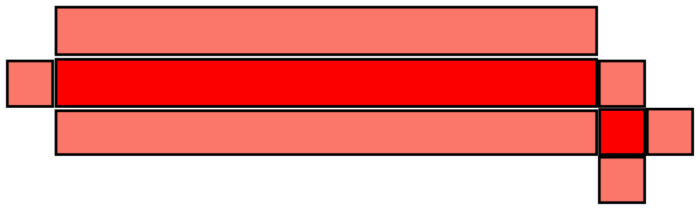




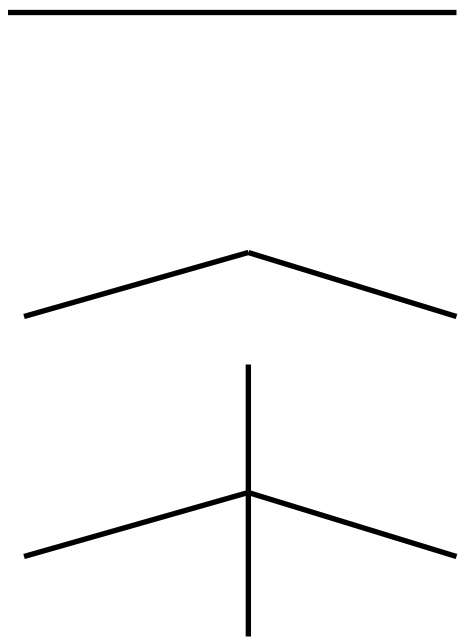
Map with terrain only

The Obstacle Mask

- Obstacle mask defines future, rather than present, obstructions
- Each square has three possible states in the obstacle mask: must be blocked, may be blocked or clear, and must be clear
- “may be blocked or clear” allows for more flexibility and randomness when placing actual obstacles



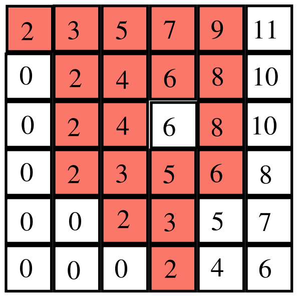
Decorative obstacles

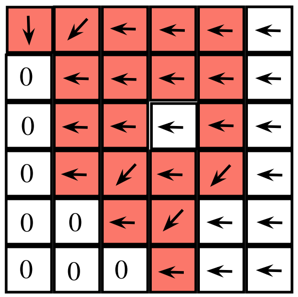


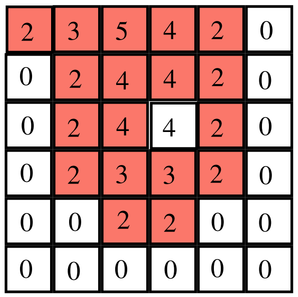
- Instead of placing decorative obstacles, the generator takes a negative approach
- Initial state is “must be blocked” for every land square, and “must be clear” for water squares
- Cut random paths, using a fractal method, through the obstruction mask
- Like the edge roughening, except that the generator adds new, perpendicular lines at each stage

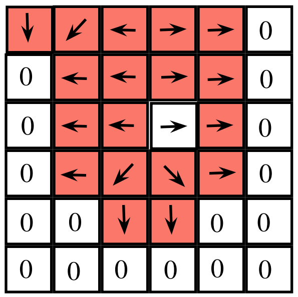
Insuring all areas are reachable

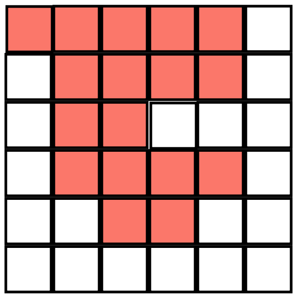
- An “A*” based pathfinding algorithm determines the distance from the first open square in the zone to all other squares in the zone.
- Cost to move orthogonally into a square is 2, cost to move diagonally is 3. Cost to move from any zero-cost cost square to any open square is zero.
- After determining distances, cut a least-cost path through the obstacle mask from any open square that is *not* connected to the main area.
- Repeat until all open squares are connected.



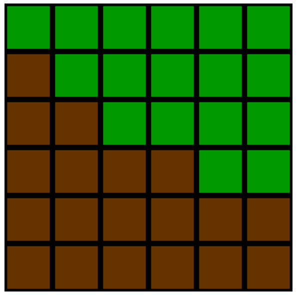




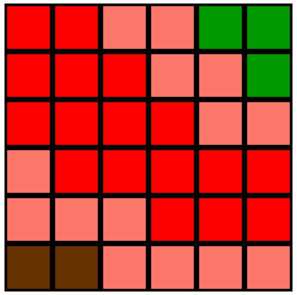




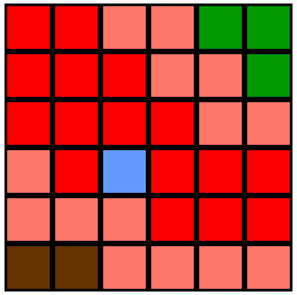
Connecting adjacent zones



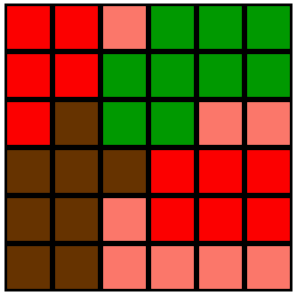
- If connected in the template with a "wide connection", stop.



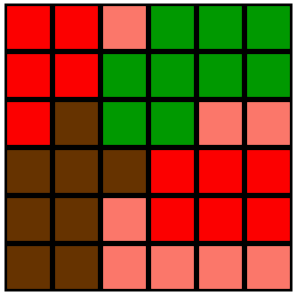
- Block border with "must be blocked" squares with adjacent "may be blocked squares".



- If not connected, stop. Otherwise select a random point along border.



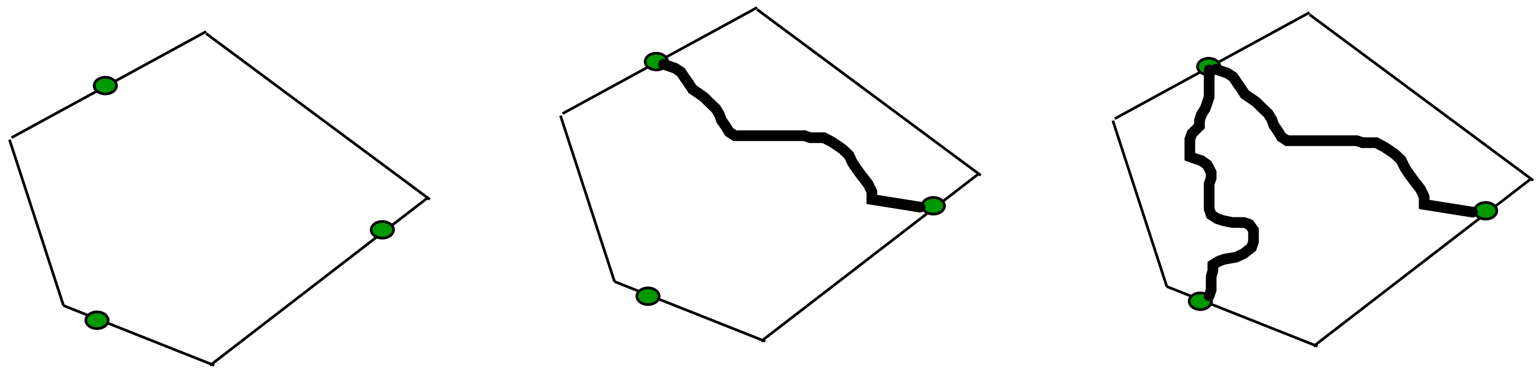
- Cut 2-wide least-cost pathway from point to each zone.



- Place guardian monster, if any.

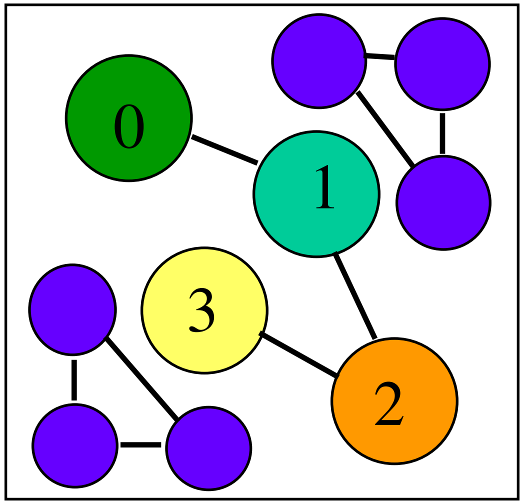
Junction Zones

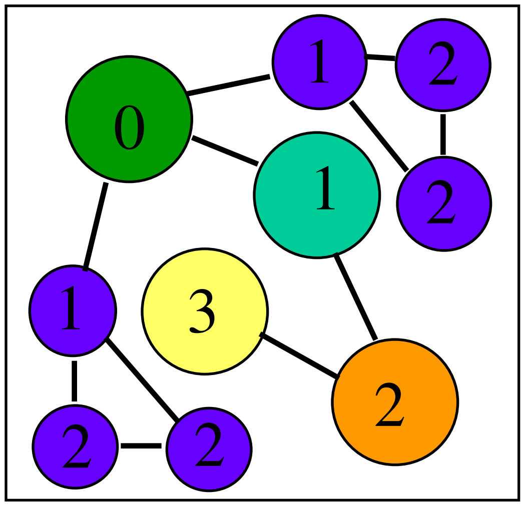
- Junction zones are intended to be narrow tunnels or passes.
- Find the points connecting this zone to adjacent zones.
- Block off entire zone.
- For each connection point, check if it's possible to reach the other connection points.
- If there is no current path between a pair of connection points, clear a line from the first connection point to the second. Randomize this line fractally.

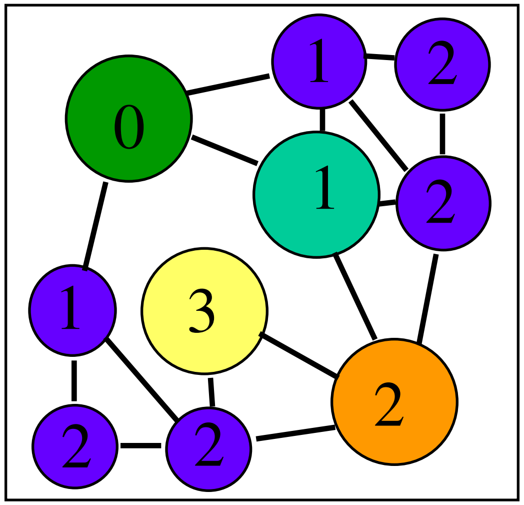


Water Connections

- Connect adjacent water zones
- For each zone, if connecting it to an adjacent water zone does not result in a shorter route to some other zone, allow it to connect to the water zone.

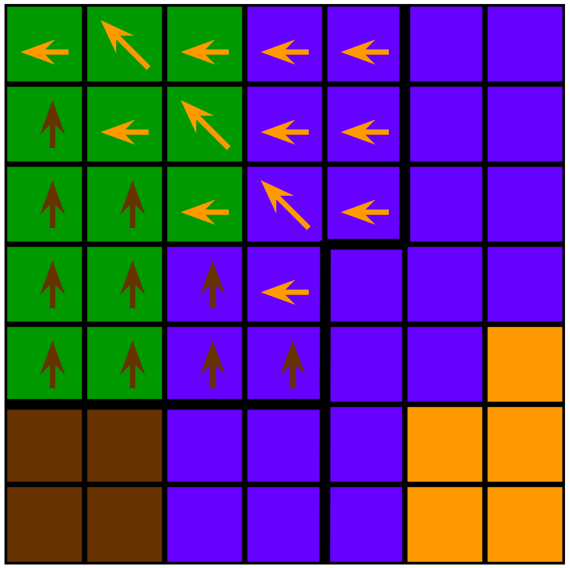






Building Shipyards

- Mark distance and direction from each square to nearest adjacent zone.
- Pick a square at random which can hold a shipyard and has a connected zone as the nearest adjacent zone.
- Mark all connected water squares, so that each zone has only one shipyard per body of water.

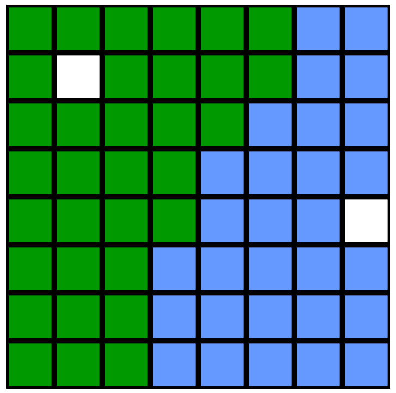


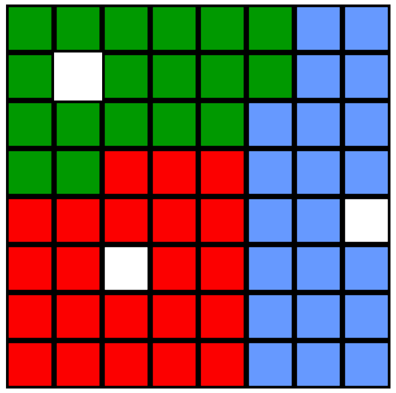
Underground Gates and Teleporters

- For underground gates, choose a random spot that overlaps in the surface and underground levels.
- If a connection is not possible via land, shipyard, or underground gate, use a teleportation gate.

Placing Objects

- Distribute objects evenly.
- Maintain a map with distance to the nearest object (town, mine, or treasure).
- After placing each new object - town, mine, or treasure - mark the map with new distances.
- Place each new object as far from all previous objects as possible.





Object Density: Concept

- Zones can vary greatly in size.
- Most objects must vary in number depending on the size of the zone.
- Towns and mines have minimum numbers, and densities for additional objects. Treasures have only densities.
- Objects are placed by type: Towns, then mines, and finally treasures.

Object Density: Implementation

- Within a type, there can be several different densities.
- Relative density within the type determines order and frequency of placement.
- Total density of all objects determines minimum distance between objects. Formula is Distance = sqrt(area / density).
- Continue to place objects until new object must be placed inside the minimum distance from another object.

Town Placement

- Zones have a minimum number of towns and castles, and a density.
- Place first town for all zones as close to center of each zone as possible.
- Towns after the first use normal object placement.

Mine Placement

- Each zone has a minimum number of mines of each type, and a density.
- Place first wood and first ore mine in a player starting zone within 12 squares of starting town, if possible. If not possible, place them as closely as possible.
- Place additional mines normally.

Monster Strength

- Treasures, mines, and sometimes connections between zones need guards.
- Monster strength depends on the value of the reward, and monster strength rating: none, very weak, weak, average, tough, or very tough.
- Zones can specify “no”, “weak”, “average”, or “tough” monsters for treasures and mines.
- Connections between zones can specify the value of the connection, which is always guarded by “average” strength monsters.
- Player selection can change the strength of all monsters up or down one category.

Selecting Monsters

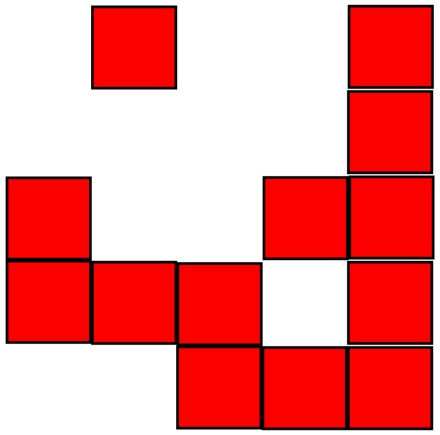
- Template specifies which alignments are legal for monsters in each zone.
- Desired number of monsters = guard value / value of individual monster.
- Discard monsters for which quantity is less than the normal average for that monster.
- Discard monsters for which quantity is greater than 100.

Treasures

- Any object which rewards the player is a “treasure” object.
- After placing towns, create a table of potential treasures, including values and relative frequency.
- Value of treasure may depend on map conditions. Creature dwellings are much more valuable if map has corresponding towns.
- Some treasures have several variations. Pandora’s Boxes may contain experience, gold, creatures, or spells.

Treasure Blocks

- Zones specify treasures in terms of a range of total value, and a density.
- Each zone can have 3 different sets of treasure value / density specs.
- Treasure blocks may contain any number of treasures, a guarding monster or border guard, and an obstacle mask.

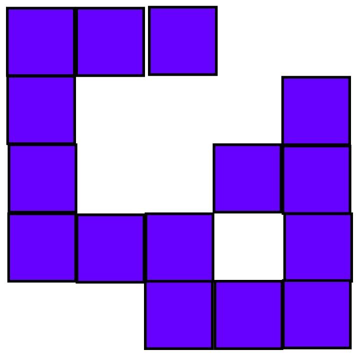
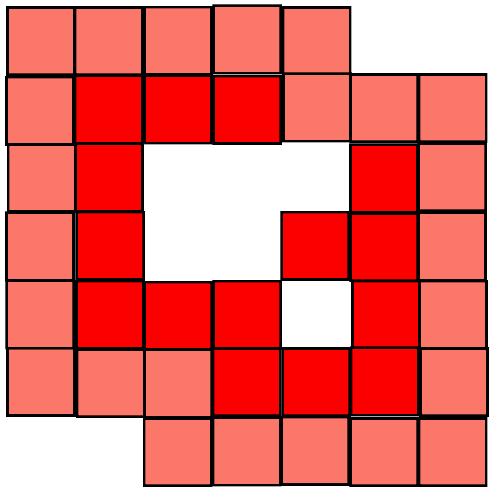
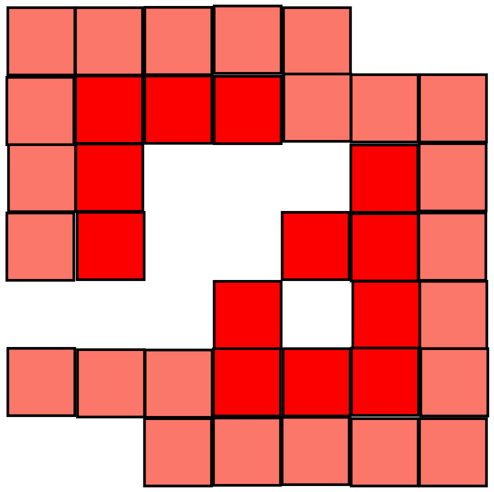


Creating a Treasure Block

- Total desired value of each treasure block is random within a range.
- Choose treasures from those worth 1/4 of unused value to all of remaining value.
- Must be able to trace a path from any treasure’s trigger point to any other inside the block.
- First treasure in block can be a permanent object.
- Additional treasures must be placed adjacent to an earlier trigger point.
- Remaining treasures must be removable.
- Many objects can only be approached from the south. New treasures cannot block or be blocked by earlier treasures.
- Some treasure objects have restricted quantities. A second Temple in the same zone is not valuable, and can appear repetitive.

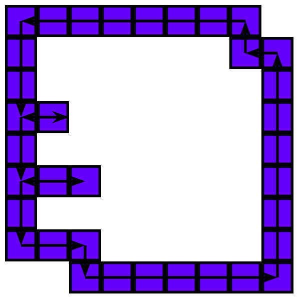
Guarding a Treasure Block

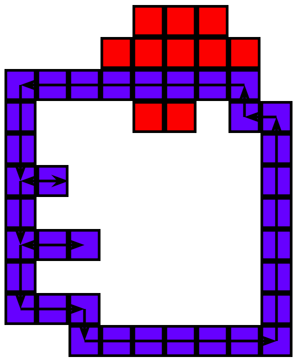
- Build list of points on perimeter of treasure block.
- Block all points which can reach treasures. Mark adjacent squares as "may be blocked."
- Choose location for monster from perimeter list.
- Clear blocks adjacent to monster.

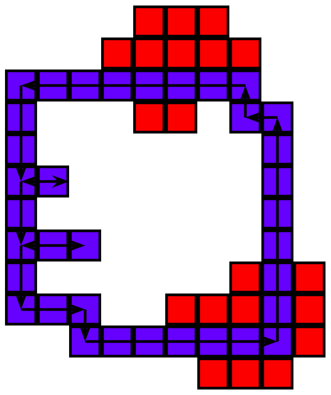




Placing a Treasure Block

- Blockage in obstacle mask can overlap previous marks in map's obstacle mask.
- To avoid sealing off areas, the treasure block cannot connect two obstacles.
- Check perimeter of block. If there is more than one transition to blocked from clear, there are two adjacent obstacles.







Artifact Quests

- Artifact quests require two objects: the artifact which is the goal of the quest, and the Seer's Hut which provides the quest.
- Select a low level artifact at random as the goal. Maintain a list to avoid duplicating these artifacts.
- Place the goal of the quest in the treasure block. Treat it as a regular treasure with an unusually high value.
- After treasure block is placed, attempt to place the Seer's Hut in another zone, preferably one 2 zones distant.

Key Quests

- Key quests include a border guard blocking access to a treasure block, and a tent to provide the key.
- Place the tent as a regular treasure in the treasure block, with a value depending on the quest.
- After the block is placed, attempt to place the second treasure block.
- Use the monster placement logic for the border guard.
- If the second block cannot be placed, remove the tent from the first block, and find an appropriate value treasure to replace it.
- It is possible for key quests to cascade.

Obstacle Selection

- Begin by selecting a square that must be blocked that is currently open.
- Examine all possible obstacles that could fill the square.
- Every obstacle type (snow covered pine tree, grassy lake, desert mountain, etc.) has a table defining aesthetics of placing that obstacle.
- Some obstacles are prohibited from appearing on some terrain types (I.e. cactus on snow).

Obstacle Selection continued

- Examine all possible ways to place each obstacle without blocking a square which must be clear.
- Calculate a weight for each obstacle and placement combination.
- Choose randomly from the weighted combinations.
- Examine squares near newly placed obstacle that must be blocked and are currently clear.

Obstacle Weighting

- Weight is based on what objects (obstacles and some items such as mines) are adjacent and what objects the obstacle will overlap.
- Some combinations are prohibited.
- Acceptable but unrelated combinations have a small weight.
- Related combinations (I.e. pine trees and snow covered pine trees) have a moderate weight.
- Identical obstacle types have a very high weight.

Obstacle Weighting Continued

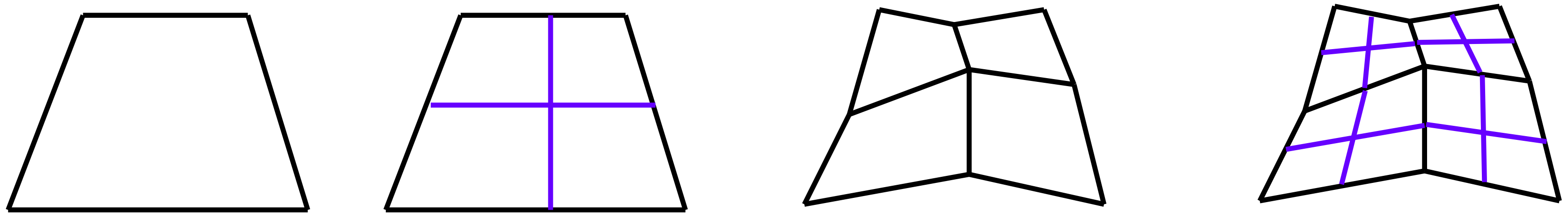
- Some layering effects, such as trees over mountains, have a moderate weight.
- Weight is also based on what terrain types the object overlaps. Some terrains have negative weights or are prohibited.
- Weighting strongly favors placing appropriate mountains near mines and forests near sawmills.

Decorative Islands

- Open expanses of water appear empty, even with objects such as flotsam and shipwrecks.
- Decorative islands add visual interest.
- To generate a decorative island, generate a fractal height map, and at every point that has positive height, place terrain.
- Entire island is marked as obstructed.

Fractal Height Maps

- Begin with a rectangle.
- Subdivide rectangle into 4 rectangles.
- Randomly displace center and 4 midpoints.
- Repeat on smaller rectangles.



Roads

- Draw roads from each town to every other town or shipyard.
- Use a pathfinding algorithm to find the shortest distance from each town to each destination.
- Pathfinding algorithm can use teleportation gates and underground gates.
- Treat existing roads as being 1/10th cost.

Rivers

- Watermills need a river with a source and a sink to look correct.
- Use a pathfinding algorithm to find closest water source and water sink.
- Randomize movement costs slightly to induce bends in the river.
- Mountains and lakes are water sources.
- Map edges and shorelines are water sinks.
- A watermill that is linked to a water sink is also a water sink.

Questions?