

Semantic Visual Search: Visual Exploration of Spore Creations



CS294 Final Project Presentation
Arpad Kovacs
2010.05.05

Problem Description



Spore database:
140 million+
creations

How do you
browse / search
this data?

How do creatures
relate to each
other?

Existing Keyword-based Search

Slow, not very interactive

Type in search terms, select category, pages of results

Inaccurate, imprecise

Mislabeled / mistagged creations, ambiguity

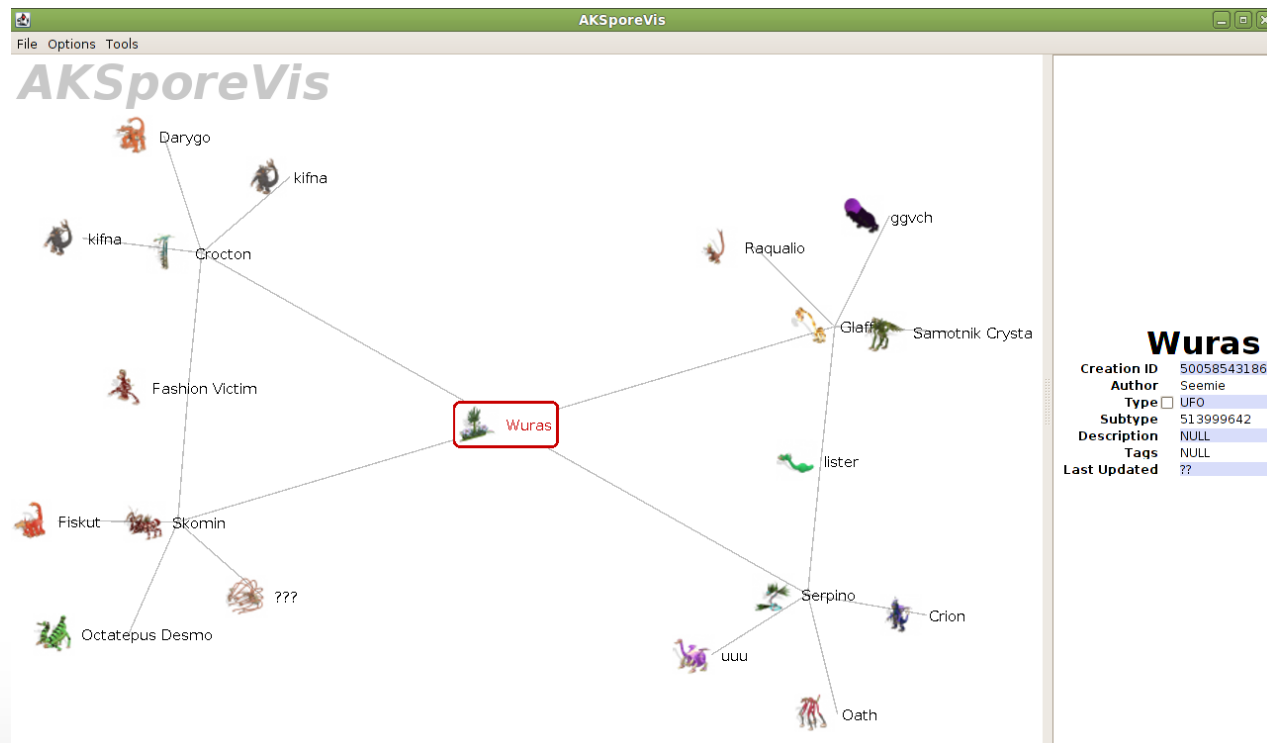
The image shows a search interface with a green and blue color scheme. At the top, there are navigation tabs: 'Home', 'Browse', and 'Search' (which is highlighted). Below the tabs is a search bar with the placeholder text 'Enter Search Term' and a 'Search' button. Under the search bar, there are filter options for 'SEARCH BY:' with checkboxes for 'Creation Name', 'Creator Name', 'Tags', and 'Description'. Below that, there are filter options for 'CREATION TYPE:' with checkboxes for 'Creatures', 'Buildings', 'Vehicles', and 'Adventures'. Each of these has a link to 'select subtypes'. At the bottom, there are filter options for 'FILTER BY:' with radio buttons for 'Newest', 'Highly Rated' (which is selected), 'Recent Highly Rated', and 'All (slow)'. There are also links for 'Select All' and 'Select None'.

Solution: Morphology Tree

Find most similar creations, link them together
semantically-meaningful results linked by morphology

Dynamic query techniques

Trees with overview, zoom & filter, details-on-demand

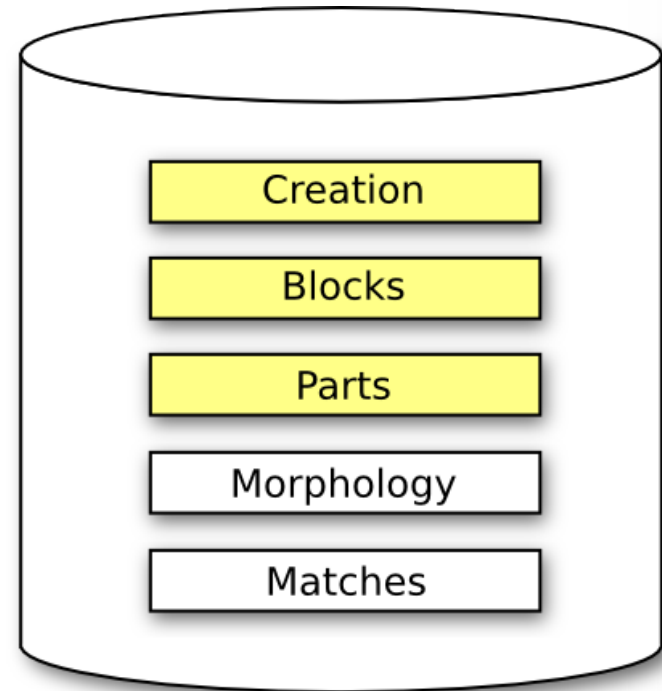


Step 1: Data Parsing

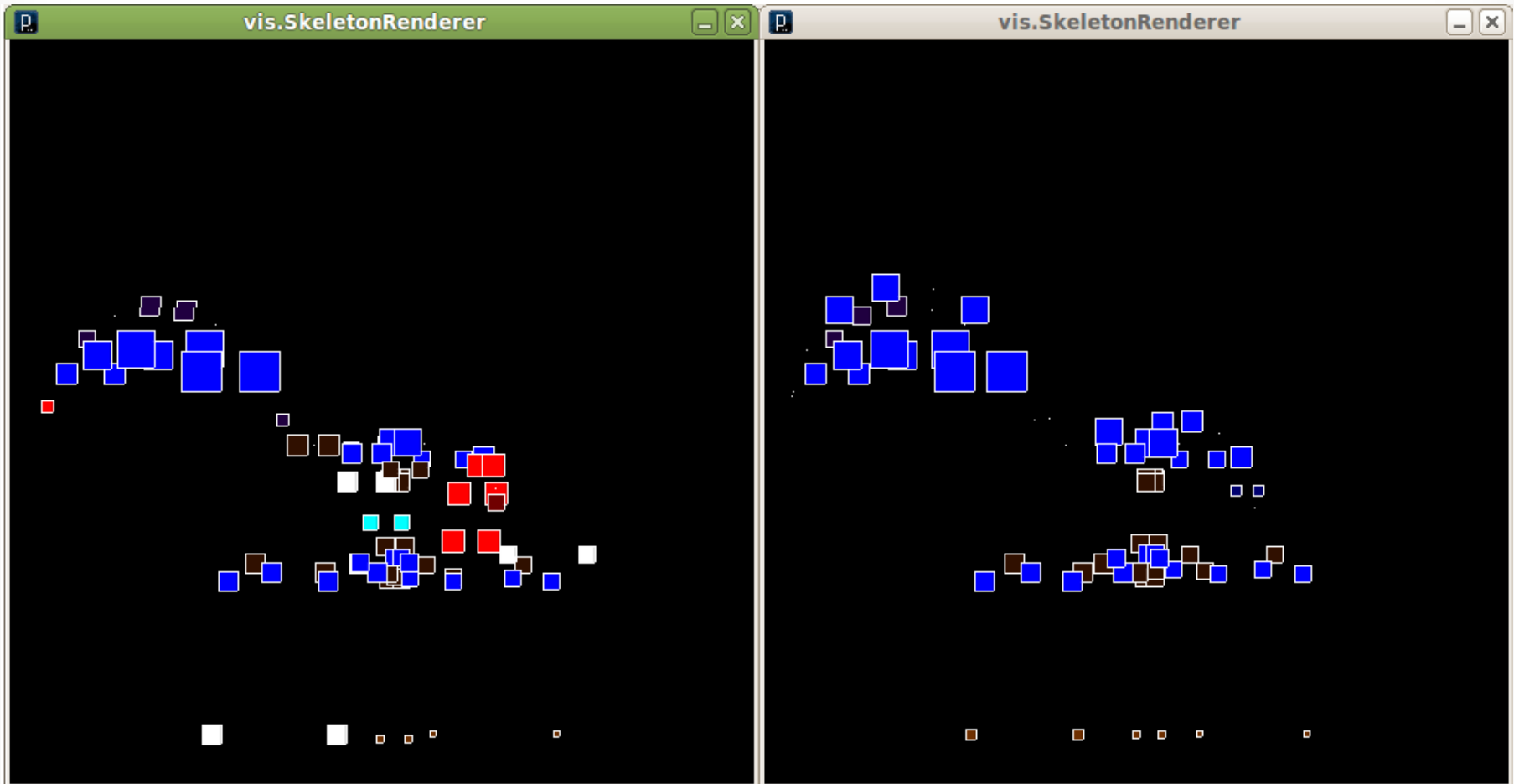
1,000 XML Files

PostgreSQL Database

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <sporemodel>
3   <formatversion>16</formatversion>
4   <properties>
5     <modeltype>0x9ea3031a</modeltype>
6     <skineffect1>0xf9081c4e</skineffect1>
7     <skineffect2>0xf726d097</skineffect2>
8     <skineffect3>0x91c84cb2</skineffect3>
9     <skineffectseed1>1234</skineffectseed1>
10    <skineffectseed2>1234</skineffectseed2>
11    <skineffectseed3>1234</skineffectseed3>
12    <skincolor1>0.961000,0.898000,0.796998</skincolor1>
13    <skincolor2>0.576469,0.572548,0.498039</skincolor2>
14    <skincolor3>1,0.941176,0.831371</skincolor3>
15    <zcorpsscore>0</zcorpsscore>
16  </properties>
17  <blocks count="88">
18    <blockref>
19      <blockid>0x40626000, 0xd05c53a3</blockid>
20      <transform>
21        <scale>0.288000</scale>
22        <position>0,-0.256773,1.672280</position>
23        <triangledirection>0,0.994836,-0.101489</triangledirection>
24        <trianglepickorigin>0,0,0</trianglepickorigin>
25        <orientation>
26          <row0>1.000000,0,0</row0>
27          <row1>0,0.994836,-0.101489</row1>
28          <row2>0,0.101489,0.994836</row2>
29        </orientation>
30      </transform>
31      <snapped>>false</snapped>
32      <childlist count="2">
33        <childid>58</childid>
34        <childid>1</childid>
35      </childlist>
36    </blockref>
37    <blockref>
38      <blockid>0x40626000, 0xd05c53a3</blockid>
39      <transform>
40        <scale>0.286555</scale>
41        <position>0,-0.158906,1.639035</position>
42        <triangledirection>0,0.850822,-0.525452</triangledirection>
43        <trianglepickorigin>0,0,0</trianglepickorigin>
44        <orientation>
45          <row0>1.000000,0,0</row0>
46          <row1>0,0.850822,-0.525452</row1>
47          <row2>0,0.525452,0.850822</row2>
```

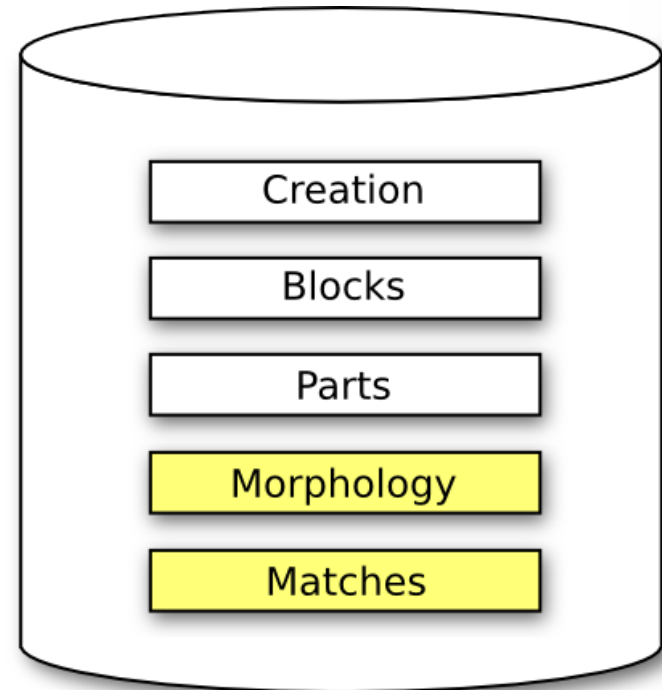


Step 2: Determine Morphology



Step 3: Calculate Matches

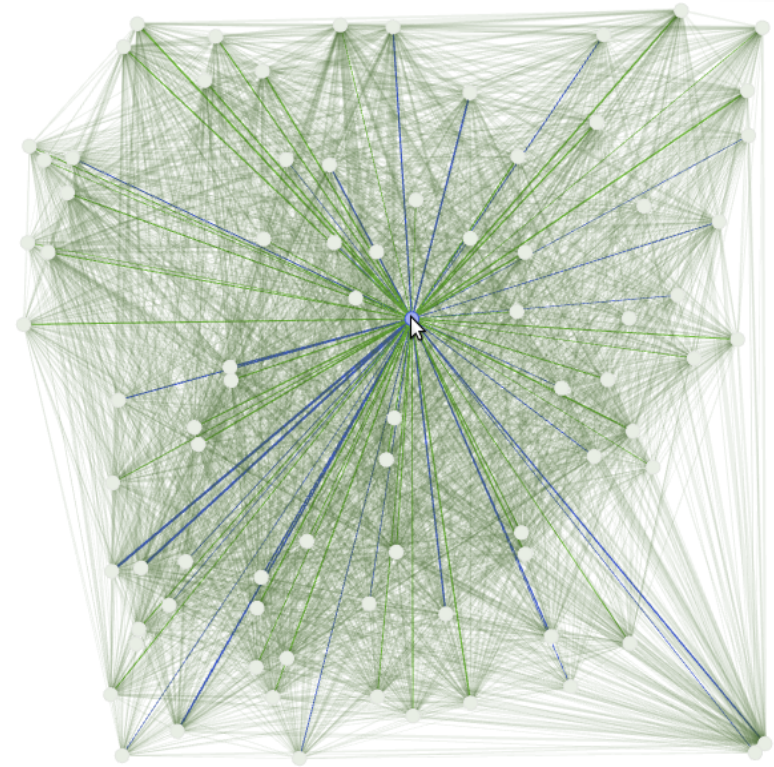
1. Find starting block of creature description
2. Generate creature topology (how parts are connected)
3. Find all sequences of 3 connected blocks in creation.
4. Generate match index for each creature in database
5. Sort creature pairs by number of matches.



Step 4: Build Graph

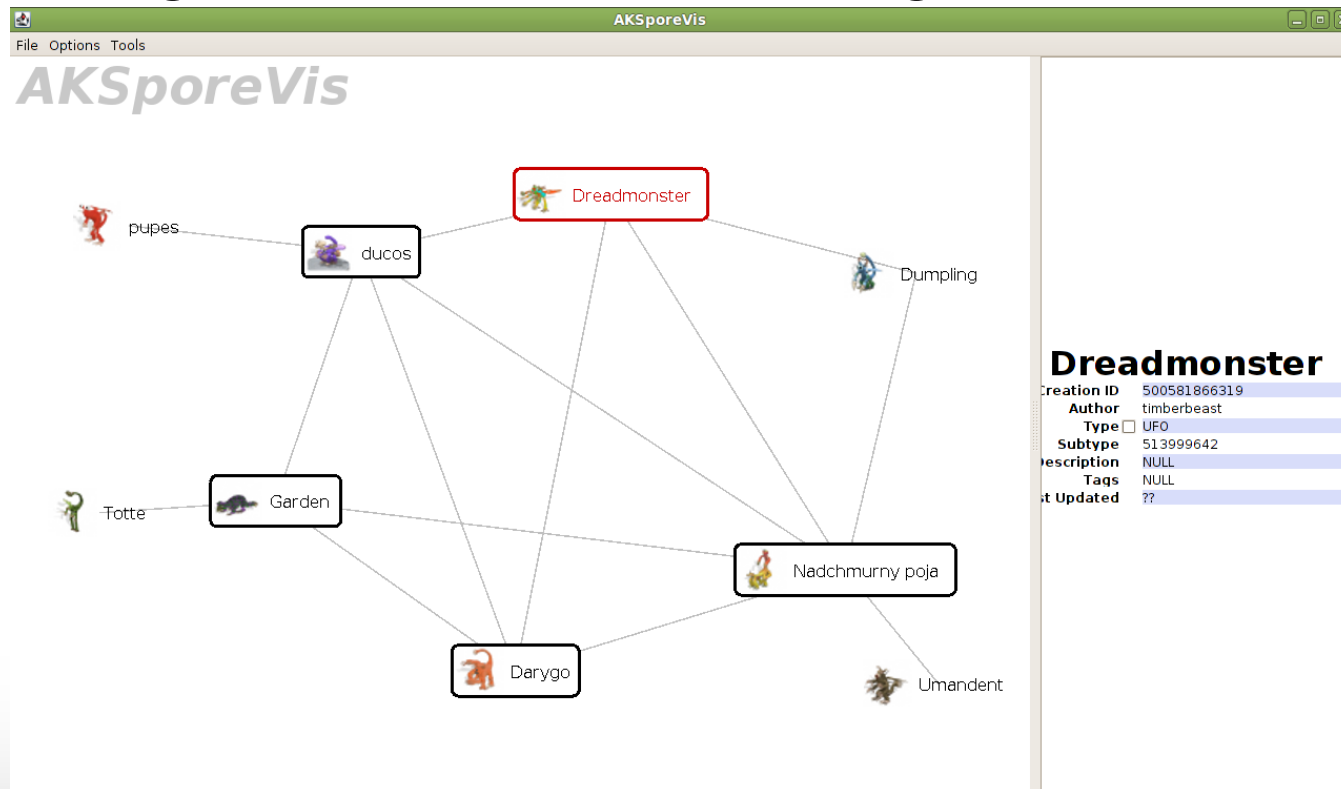
1. Start with 80 creatures, calculate match index between each pair.
2. Initial graph is too dense (3000 edges), does not convey useful information
3. Solution: Reduce number of nodes, only connect a creature to its 4 top matches

Gephi Demo



Step 5: Build Visualization

1. Build visualization based on Prefuse framework; interactive zoom, details on demand, animation
2. Use small graph and display pictures of creatures
3. Edges are still annoying; Solution: turn graph into tree using hierarchical clustering



Step 6: Iterate and Refine

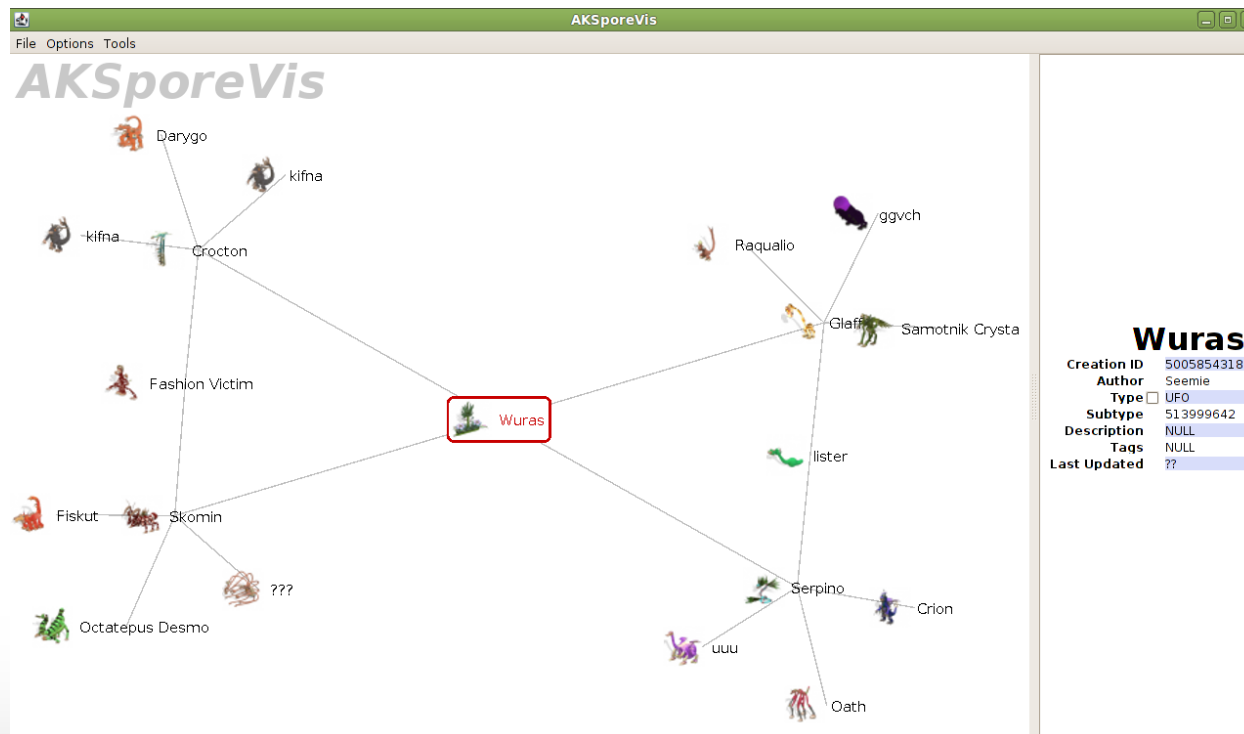
1. Increase size of dataset, add additional metrics for determining matches
2. Improve performance via caching most recently used images and data.
3. Add additional functionality, search criteria.

The screenshot shows the AKSporeVis application interface. The main window displays a network graph with 'Darygo' as the central node, highlighted with a red border. It is connected to several other nodes, each with a small image icon and a text label: 'pupes', 'Moruche', 'Dreadmonster', 'Drakion', 'giganotsarus', 'Mightyfenex', 'Garden', 'Oglaw', 'Pio', 'Nadchmurny poja', 'Tuolias', 'Totte', 'Spineulator', 'ducos', and 'assddff'. On the right side, a metadata panel for 'Darygo' is displayed, showing the following information:

Creation ID	500581867479
Author	Grefool
Type	<input type="checkbox"/> UFO
Subtype	513999642
Description	NULL
Tags	NULL
Last Updated	??

Future Work

1. Better visualization of specific matching parts.
2. Find more optimal node layout (currently using Fruchterman-Reingold algorithm)
3. Add additional functionality, search and filtering criteria.



Questions?

Image Credits and Attribution

Spore

<http://www.spore.com>

Prefuse

<http://prefuse.org>

Vizster

<http://hci.stanford.edu/jheer/projects/vizster/>

Full list of references at:

<http://vis.berkeley.edu/courses/cs294-10-sp10/wiki/index.php/FP-ArpadKovacs>