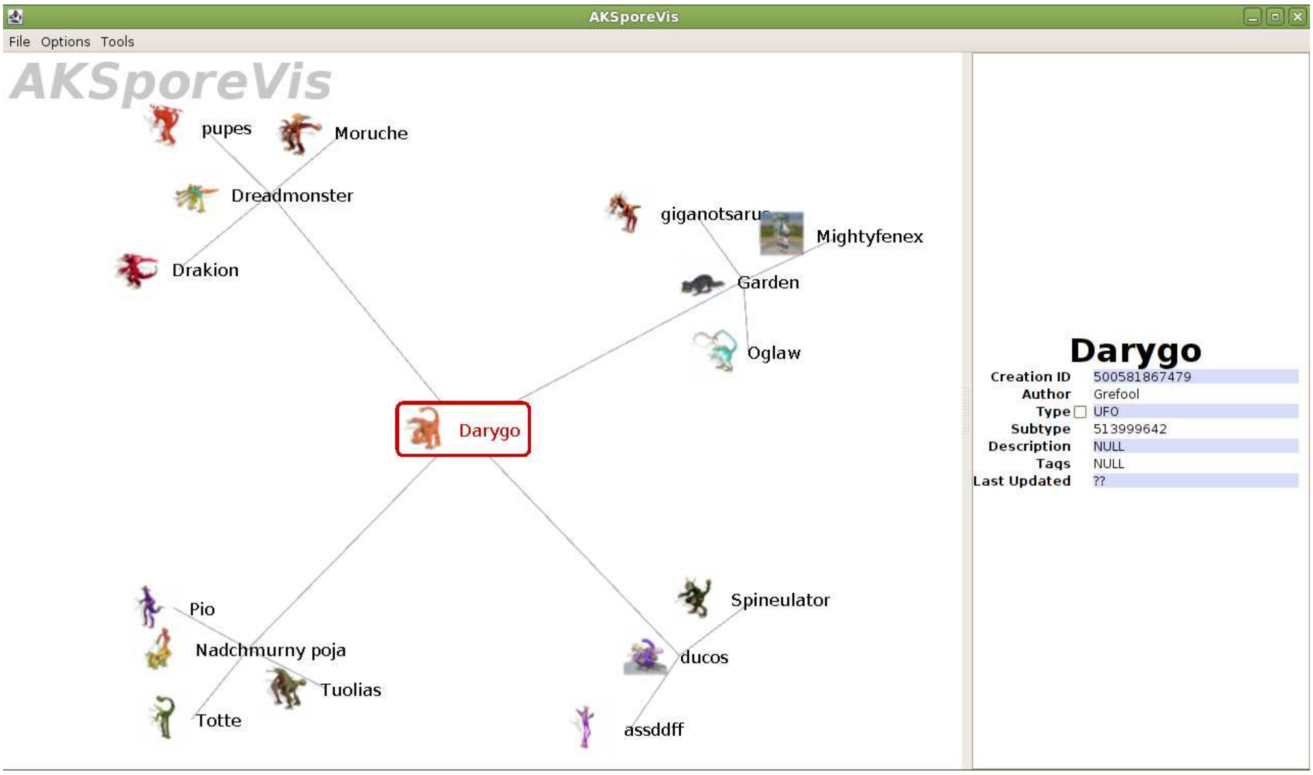
SporeVis: Interactive Exploration of Spore Creations by Morphology

Arpad Kovacs[*]
University of California, Berkeley



Darygo

Creation ID	500581867479
Author	Grefool
Type □	UFO
Subtype	513999642
Description	NULL
Tags	NULL
Last Updated	??

Figure 1: Screenshot of the SporeVis interface. The left panel displays a tree, where the leaves consist of Spore creatures and their names, and the edges connect each creature to its 4 closest neighbors by morphology. The right panel shows additional details about the selected creature.

Abstract

This paper presents a novel system for classifying, organizing, and visualizing creatures from the 2008 video game Spore according to morphology. The creatures are decomposed into a sequences of their constituent parts using an innovative encoding scheme that preserves both the hierarchical and semantic structure of the creation. At the same time, this encoding scheme is optimized for fast lookup of individual blocks and rapid comparison of sequences between dissimilar creatures. The system groups creatures into clusters according to the degree to which their block sequences match, and allows the user to perform real time dynamic queries to filter, view, and compare details of individual or multiple creatures on demand. The techniques described in this paper may be applied to any dataset composed of assemblies consisting of building blocks that have varying levels of parts commonality.

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces

Keywords: Visualization, graph, tree, morphology, phylogeny, spore, interactive, dynamic queries

1 Introduction

Modern video game publishers increasingly rely on user-generated content to add variety and increase customer loyalty to their games. Electronic Arts' 2008 video game Spore takes this trend to a new level by turning the ability to generate creations with arbitrary appearance, structure, and functionality into a central feature of the gameplay. However, as the set of available Spore creatures has grown to over 140 million creations as of May 2010, several issues have cropped up due to the sheer popularity of the user-generated content system and its ability to create an almost unlimited combination of creations.

Browsing through and searching for a creature with a particular shape or morphology in this vast datasource has become all but impossible. Creatures in the Spore database may be given a name, description, and are sometimes marked with short phrases or tags; however in practice few authors tend to take the time to properly annotate their creations before uploading them to the database. Consequently, the keyword-based search on Sporepedia, the main repository of Spore creations, yields many irrelevant and often unre-

[*] e-mail: akovacs@berkeley.edu

lated results for a given search term, while occasionally completely ignoring applicable creations that may have been mislabeled. The available filtering criteria ("Newest", "Highly Rated", "Recent Highly Rated", "Featured", "All") are primarily focused on freshness and users' ratings, and therefore do not help narrow the results down according to morphology. As a result, users of Sporepedia are often forced to page through hundreds of results before they can find a creature that matches their specific interests. This results in a large gulf of execution between the user's goals of finding a specific creature, and the required steps to successfully perform the desired action of finding a suitable match [Norman and Draper 1986].

In addition, the current Sporepedia search interface does not provide the user with any notion of how creatures are related to each other. The results are arranged in a grid and ordered by upload time, which is not a semantically meaningful criterion. This presents an opportunity for a new system for browsing and searching through user-generated content that visualizes parallel evolution and cross-pollination of creation construction techniques in a semantically-meaningful and morphology-aware manner. In particular, the proposed system should facilitate the exploration of this interesting dataset through dynamic queries using the overview, zoom & filter, details-on-demand paradigm advocated by the Visual Information-Seeking Mantra [Ahlberg and Shneiderman 1994].

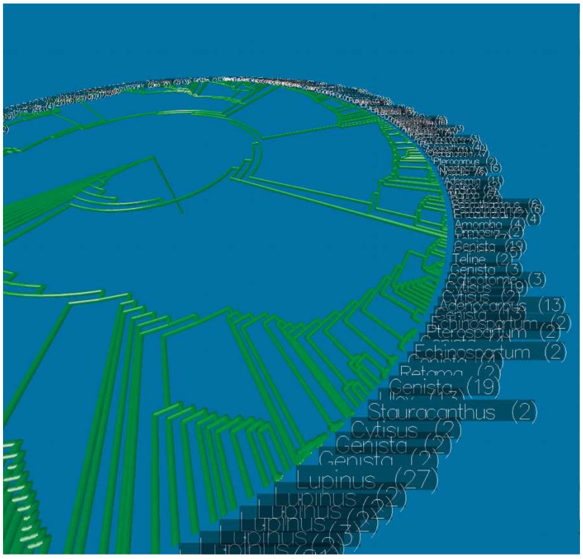
2 Related Work

While there has been no work done to date on visualizing Spore creations by morphology, the proposed system shares some elements in common with earlier developments in the biological sciences that seek to build evolutionary trees. In particular, research in the field of computational phylogenetics aims to assemble phylogenetic (evolutionary) trees and networks representing hypothesis about the evolutionary ancestry of a series of taxa based on inherited characteristics and morphological analysis [Semple and Steel 2003]. The system is related to earlier works in computer science that attempt to visualize large hierarchical tree and graph structures (often over a given time duration), including systems such as GraphViz [Gansner and North 1999], GEVOL [Collberg et al. 2003], and Vizster [Heer and Boyd 2005]. Finally, SporeVis' emphasis on interactive dynamic queries has many parallels to the Dynamic Homefinder [Williamson and Shneiderman 1992] and Spotfire [Ahlberg and Shneiderman 1994] systems, as well as more recent works such as Generalized Selection via Interactive Query Relaxation [Heer et al. 2008].

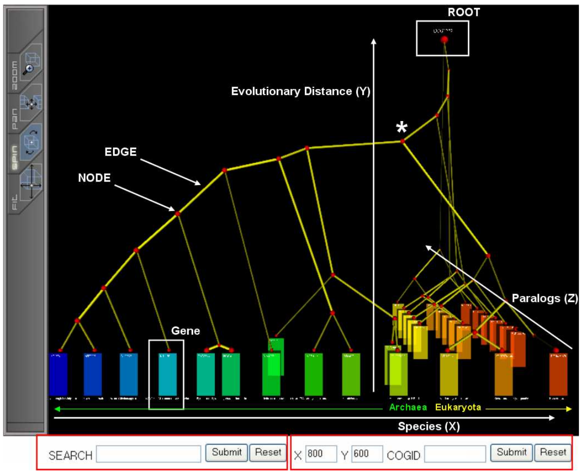
Although biologists have been creating static phylogenetic trees since before the time of Darwin [Archibald 2009], it is only recently that computerized techniques have been applied to programatically generate hierarchical groupings of taxa. The NSF Tree of Life Web Project Initiative [Maddison et al. 2007], an early attempt at interactive phylogenetic trees, consists of hierarchically linked HTML pages, in the form of a cladistically organized branching evolutionary tree of life. Each page contains information about one particular group of organisms organized as a branched tree, thus showing hypothetical relationships between different groups of organisms.

Various tools such as Web Tree View [Zhai et al. 2002], Java Tree View [Saldanha 2004], ATV [Zmasek and Eddy 2001], and TaxonTree [Parr et al. 2004] have been built for viewing and annotating pre-built phylogenetic trees. However, these tools do not automatically filter and cluster the results based on user-provided criteria and only allow for limited interactivity, therefore I classify them as static rather that truly dynamic and interactive visualizations.

A more recent phylogenetic tree visualization system is the Paloverde OpenGL phylogeny browser [Sanderson 2006]. Paloverde aims to provide phylogeneticists with rapid navigation of binary trees' clades, without losing sight of the leaf names and maintaining the tree representation in Euclidian space. The authors extoll the use of realistic lighting, perspective, and other attributes as features which add "realism", however in the eyes of many critics, the gratuitous use of 3D detracts from the visualization by diverting the user's attention from the relevant data.



Another OpenGL-based phylogenetic tree visualization system is the 3D Phylogenetic Tree Explorer [Kim and Lee 2007]. This system projects phylogenetic trees onto three spatial axes, with leaves assigned such that species are catagorized on the X-axis while paralogs (genes related by duplication within a genome) are represented on the Z axis; the Y coordinate represents evolutionary distance, just as in conventional 2D trees.



The difference between these existing computerized graph visualizations and my proposed system is the data domain, as well as the deployment of interactive, direct-manipulation querying and filtering techniques. In addition, the existing systems are tailored for a technical, scientifically literate audience, while my system is intended for use by people with little or no background knowledge of biology.

In the realm of Spore visualizations, the main reference sources of Spore creations are Electronic Art's official Sporepedia encyclopedia [Spo a], and the Sporistics analytics service [Spo b]. Although some studies have been performed on ranking Spore creations and analyzing community interactions [Twardos 2009], so far there have been no previous works that focus on the morphology of the Spore creations themselves.

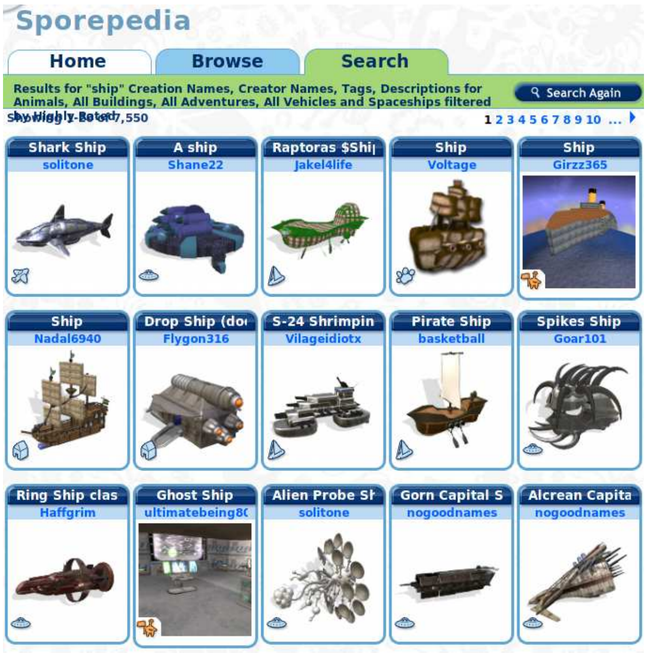
3 Visualization Design

3.1 Design Goals

My goal for SporeVis was to build a visualization system that users of the Spore Creature Creator could use to facilitate discovery of creations with specific morphologies, and understand how their own creations compare to existing creations. I wanted to support the open, collaborative, and exploratory nature of the Spore Creation ecosystem while giving users easier access to search and group patterns.

3.2 Existing Sporepedia Interface

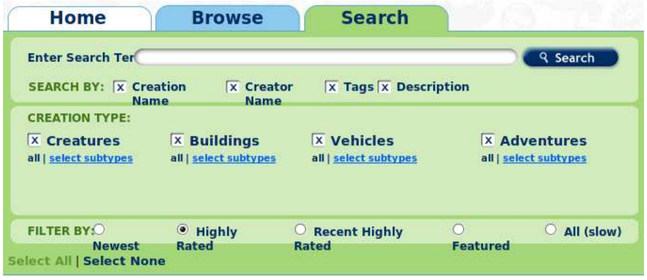
The existing Sporepedia tool allows for browsing multiple creations at a time, with rectangular icons showing each creature's name, author, and a small image of the creature arranged in a 4x5 grid, showing 20 creature results per page. Unfortunately, the linear, paginated format of this interface limits interactivity and does not provide a coherent organization of the available creations. In addition, although it is possible to view detailed information for a specific creature, this involves the appearance of an intrusive popup which covers a large portion of the screen, and hides information for all but the selected creation.



Results for "ship" Creation Names, Creator Names, Tags, Descriptions for Animals, All Buildings, All Adventures, All Vehicles and Spaceships filtered by Highly-Rated
Showing 1-20 of 7,550

Shark Ship – solitone
A ship – Shane22
Raptoras $Shi| – Jakel4life
Ship – Voltage
Ship – Girzz365
Ship – Nadal6940
Drop Ship (do – Flygon316
S-24 Shrimpin – Vilageidiotx
Pirate Ship – basketball
Spikes Ship – Goar101
Ring Ship clas – Haffgrim
Ghost Ship – ultimatebeing8(
Alien Probe Sh – solitone
Gorn Capital S – nogoodnames
Alcrean Capita – nogoodnames

Another flaw of Sporepedia is the keyword-oriented search interface, which relies very heavily on the author proper labeling his/her creation's name, description, and providing a list of salient tags. This is a labor-intensive (on the part of the author), inefficient, and often inaccurate approach, since many creations are not given descriptions or tags at all, or even worse mislabeled, which causes them to be categorized incorrectly. Finally, it there are little or no options available for refining queries. The available filtering criteria ("Newest", "Highly Rated", "Recent Highly Rated", "Featured", "All") are primarily focused on freshness and users' ratings, and therefore do not help narrow the results down according to morphology. As a result, users of Sporepedia are often forced to page through hundreds of results before they can find a creature that matches their specific interests, and even at that point it is hard to find additional similar results.



Enter Search Ter

SEARCH BY: ☒ Creation Name ☒ Creator Name ☒ Tags ☒ Description

CREATION TYPE:
☒ Creatures all | select subtypes
☒ Buildings all | select subtypes
☒ Vehicles all | select subtypes
☒ Adventures all | select subtypes

FILTER BY: ○ Newest ● Highly Rated ○ Recent Highly Rated ○ Featured ○ All (slow)

3.3 Key Modifications

The shortcomings of the existing Sporepedia interface inspired me to develop richer network overview and exploratory tools, while maintaining the colorful, kid-oriented nature of the Sporepedia interface. I realized that imagery was essential for uniquely identifying distinct creations, and therefore must play a key role in the visualization. Consequently, each creature is represented as a node containing an image of the creation which is asynchronously retrieved from the Internet, along with its name as a caption, in order to distinguish it from a sea of similar neighboring creatures.

In addition, I did not want to eliminate the existing detailed creation information view provided by Sporepedia; however I wanted to display this data in a less intrusive manner than an attention-diverting popup. As a result, I decided to create a creation details panel to right of the network display showing details of the creation such as its name, author, type, subtype, description, tags, and date at which it was uploaded. The creation details panel also provides direct-manipulation searches over key words or phrases selected by the user.

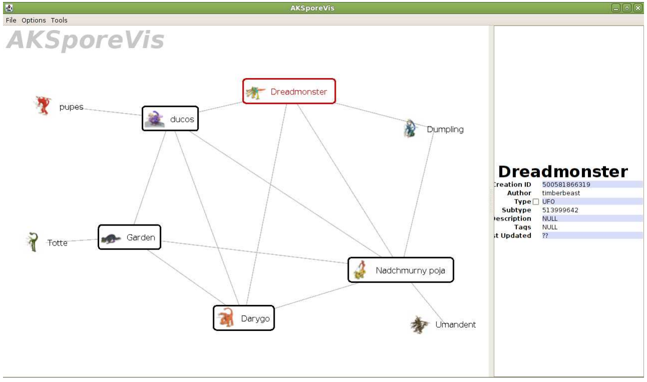
4 Visualization Implementation

4.1 Architecture

The SporeVis system is built as a Java application, primarily using the Prefuse visualization framework [Heer et al. 2005], which is designed specifically for interactivity and animation. The creations are arranged in a tree, in which the nodes are the creations, while the edges consist of connections between each creature and its 4 closest neighbors by morphology. For visualizing comparative creature morphologies and skeletal structures, I embedded Processing.org PApplets as tooltips which appear when hovering over creature nodes or edges. The entire system is backed by a PostgreSQL database that allows the user to filter the dataset and run dynamic-queries in real-time.

4.2 Layout

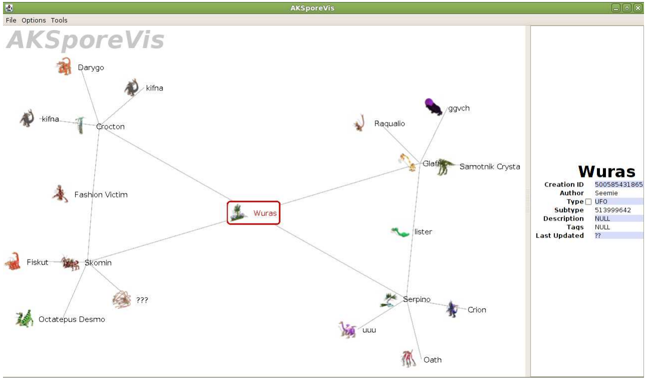
The SporeVis network layout is computed using the Fruchterman-Reingold force-directed algorithm, in which nodes repel each other and edges act as springs. Thanks to this layout, creature nodes are grouped into identifiable clusters based on their amount of connectivity. Although the connectivity between each creation to every other creation is calculated, I choose only to connect each creature with its 4 neigbors with the highest matching index. This thresholding is performed to avoid excessive connections between relatively unrelated creations, which would only serve to clutter the graph and hide the stronger relations.



Dreadmonster

Creation ID	500581866319
Author	timberbeast
Type □	UFO
Subtype	513999642
Description	NULL
Tags	NULL
Last Updated	??

There are two options for the layouting algorithm. The first option arranges nodes in a graph-like fashion, with each node connected to its 4 most similar neighbors irrespective of the overall graph structure. The second layout option provides for a more hierarchical organization by connecting each node only to the top 4 matching neighbors that are not already visible in the visualization. This generates a more aesthetically-pleasing tree structure, but is less accurate, since two creatures that may be very similar, but happen to be located on different branches of the tree are prohibited from being connected.



Wuras

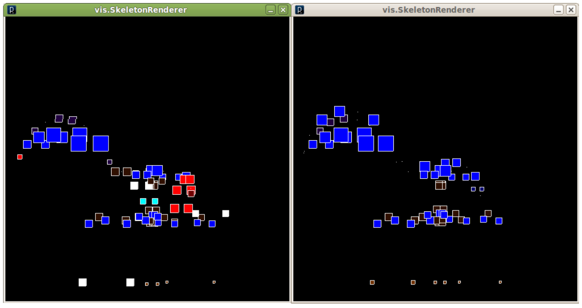
Creation ID	500585431865
Author	Seemie
Type □	UFO
Subtype	513999642
Description	NULL
Tags	NULL
Last Updated	??

4.3 Interaction

The user may interact with the SporeVis system primarily through simple mouse operations. Clicking on a creation causes the corresponding creature's salient data to appear in the profile panel to the right of the network view. The user may drag a node in order to move it around the screen; upon releasing the mouse button, the creature node is once again subject to the force-directed layout.

The SporeVis system highlights nodes based on connectivity in the larger network context. Upon hovering the mouse hovers over a creature, it causes that creation, as well as its neighboring (closest match) creations, and 2nd-degree neighbors to highlight. The highlighting is performed via a graded color scale that encodes relatedness using both intensity and hue, from red-orange at the selected creation to orange at its 1st-degree neighbors to yellow at its 2nd-degree neighbors. All other creatures in the graph are correspondingly desaturated into grayscale images, to distingush related and distant creatures.

Upon hovering over a creature node, the SporeVis system also displays a visualization of the creature's skeletal structure, in a popup tooltip. The skeletal visualization rotates, and parts are scaled by z-axis depth to provide the viewer with a 2.5D visualization that avoids the issues of part occlusion and depth misperception that would be problematic in a static visualization. The creature's parts are colored distinctly by part-type (for instance, vertebrae are blue, weapons are shown in red, etc), in order to assist the viewer in understanding the creature's morphology.



When hovering over an edge connecting two creatures, both creatures' skeleton structures are shown simultaneously, and analogous parts are outlined in yellow. As the creatures rotate simultaneously, yellow lines intermittently connect the analogous parts that are facing the viewer.

4.4 Navigation

The SporeVis system also facilitates rapidly moving across different regions of the creation morphology graph via panning and zooming functionality. By dragging the background of the display with the left mouse button down, the user may pan the screen in any direction. If the user holds down the right mouse button and then translates the mouse pointer vertically up or down the screen, then the screen zooms in or out. For performance reasons, the images are displayed in low resolution when zoomed out, and update to higher resolution photos when zoomed-in to double the normal scale.

5 Algorithm

5.1 Data Parsing

Since Electronic Arts does not allow developers to directly access the entire Spore database, the information must be retrieved through the Spore API [Moskowitz et al. 2009] in the form of XML files. In order to allow the application to perform real-time dynamic queries and work offline, I cache this data locally within a PostgreSQL database.

Please email the author at akovacsATberkeleyDOTedu for the full version of the paper which contains additional details for this section.

5.2 Generating Matches

Each creature's part topology is encoded in a tree-like fashion. The encoding scheme begins by finding the root block of the creation; this is the only block in the creature which is not a child of any other block. From this root block, the algorithm finds its children, which are all blocks that are directly linked to the root block. The blocks are encoded using a pyramidal scheme, which preserves both the hierarchical and semantic structure of the creation. At the same time, this encoding scheme is optimized for fast lookup of individual blocks and rapid comparison of sequences between dissimilar creatures. The system groups creatures into clusters according to the degree to which their block sequences match.

The hierarchically encoded blocks are arranged into sequences of 3 consecutive blocks (parent, child, grandchild). which are generated for each creature, and then are cached locally in a separate table for fast lookup. At this this point, the system runs a variant of the BLAST (Basic Local Alignment Search Tool) algorithm used in biology to find matching sequences. The challenge was to encode these connections of 3 in a way that makes it easy to traverse the tree in a hierarchical, depth-first fashion, but also enables detection of sibling parts which are at the same level. There exist Spore creatures that have a branching factor of up to 14 per level, so the encoding scheme must be robust to handle very elaborate and complex creature structures. This technique also allows the system to restrict the depth and granularity of the matches to a certain number of levels, so that matches are only counted for the main torso of the creation, rather than for insignificant features like hairs on a limb.

Please email the author at akovacsATberkeleyDOTedu for the full version of the paper which contains additional details for this section.

5.3 Tree Construction

Please email the author at akovacsATberkeleyDOTedu for the full version of this paper which contains this section.

6 Discussion

Please email the author at akovacsATberkeleyDOTedu for the full version of this paper which contains this section.

7 Future Work

Please email the author at akovacsATberkeleyDOTedu for the full version of this paper which contains this section.

Acknowledgements

The author would like to acknowledge the valuable feedback and advice provided by Professor Maneesh Agrawala, Associate Professor in Electrical Engineering and Computer Science and instructor of the CS294 Information Visualization class at the University of California, Berkeley. I would also like to thank the developers of the Prefuse Information Visualization toolkit, and Processing.org framework, as well as Jeffrey Michael Heer, the author of the Vizster online social network visualization upon which I built my own system, for making these powerful software packages available under an open-source license. Finally, I would also like to express my gratitude to the students of CS294 for their insightful feedback and suggestions on how to improve my visualization.

References

Ahlberg, C., and Shneiderman, B. 1994. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, 313–317.

Archibald, J. 2009. Edward Hitchcock's Pre-Darwinian (1840) "Tree of Life". *Journal of the History of Biology 42*, 3, 561–592.

Collberg, C., Kobourov, S., Nagra, J., Pitts, J., and Wampler, K. 2003. A system for graph-based visualization of the evolution of software. In *SoftVis '03: Proceedings of the 2003 ACM symposium on Software visualization*, ACM, New York, NY, USA, 77–ff.

Gansner, E. R., and North, S. C. 1999. An open graph visualization system and its applications to software engineering. *Software - Practice and Experience 30*, 1203–1233.

Heer, J., and Boyd, D. 2005. Vizster: Visualizing online social networks. In *INFOVIS '05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, IEEE Computer Society, Washington, DC, USA, 5.

Heer, J., Card, S., and Landay, J. 2005. Prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 421–430.

Heer, J., Agrawala, M., and Willett, W. 2008. Generalized selection via interactive query relaxation. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, 959–968.

Kim, N., and Lee, C. 2007. Three-dimensional phylogeny explorer: Distinguishing paralogs, lateral transfer, and violation of "molecular clock" assumption with 3d visualization. *BMC Bioinformatics 8*, 1, 213.

Maddison, D., Schulz, K., and Maddison, W. 2007. The tree of life web project. *Zootaxa 1668*, 19–40.

Moskowitz, D., Shodhan, S., and Twardos, M. 2009. Spore api: accessing a unique database of player creativity. In *SIGGRAPH '09: SIGGRAPH 2009: Talks*, ACM, New York, NY, USA, 1–1.

Norman, D. A., and Draper, S. W. 1986. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

Parr, C., Lee, B., Campbell, D., and Bederson, B. 2004. Visualizations for taxonomic and phylogenetic trees. *Bioinformatics 20*, 17, 2997–3004.

Saldanha, A. 2004. Java treeview–extensible visualization of microarray data. *Bioinformatics 20*, 17, 3246–3248.

Sanderson, M. 2006. Paloverde: an opengl 3d phylogeny browser. *Bioinformatics 22*, 8, 1004–1006.

Semple, C., and Steel, M. 2003. *Phylogenetics*. Oxford Univ. Press.

Twardos, M. 2009. Sporepedia: A Case Study in the Popularity Dynamics of Social Media.

Williamson, C., and Shneiderman, B. 1992. The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, New York, NY, USA, 338–346.

Zhai, Y., Tchieu, J., and Saier, M. 2002. A web-based tree view (tv) program for the visualization of phylogenetic trees. *J Mol Microbiol Biotechnol 4*, 1, 69–70.

Zmasek, C., and Eddy, S. 2001. Atv: display and manipulation of annotated phylogenetic trees. *Bioinformatics 17*, 4, 383–384.