Electronic Teaching of Algorithms

Matthew Bucci, Alec Williams, Trevor Maglione, Osama Haj, William King

Project Overview

Our tool will help students understand some graph traversal algorithms taught in Drexel's data structures courses via implementation and visualization.

Users will write code to implement the algorithms and see the visualized results.

- *Search: Depth-first and Breadth-first*
- *Shortest Path: Djikstra's and Bellman-Ford*
- ***Minimum Spanning Trees: Prim's and Kruskal's***

Purpose

Can be difficult to understand algorithms with only code, visualizations make concepts easier to comprehend

Gives the students the ability to implement algorithms and practice with code not just conceptually

Saves TA’s time on grading through static code analysis and obvious visual feedback

Stakeholder + Audience

Mark Boady ***Assistant Teaching Professor and Ph. D. Student***

Students taking the course **CS 260 - Data Structures** or a course with similar content.

What we presented previously

Project setup as local program using Kivy

Graphics and code interpreting were ready

- Bugged during the live demo due to resolution

Changes in the last quarter

Completely new project architecture

- Now a web app on flask, python backend, nodejs

Lowered scope to ensure minimal viable product

- Only one type of problem implemented
- No new visuals

Difficulties we faced

- Team restructuring
  - Happened two weeks after new development cycle started, critical timing
- Our management software locked us out
  - Couldn’t even check the tasks we had set up
- Time schedule was much tighter since we needed to spend more time on development
  - Not as much time for testing and polishing

Testing

Static Code Analysis

- Tedious, easy to setup but Flask renders many false positives due to project structure

Test Cases

- Web system creates new challenges

Demo

Thanks!