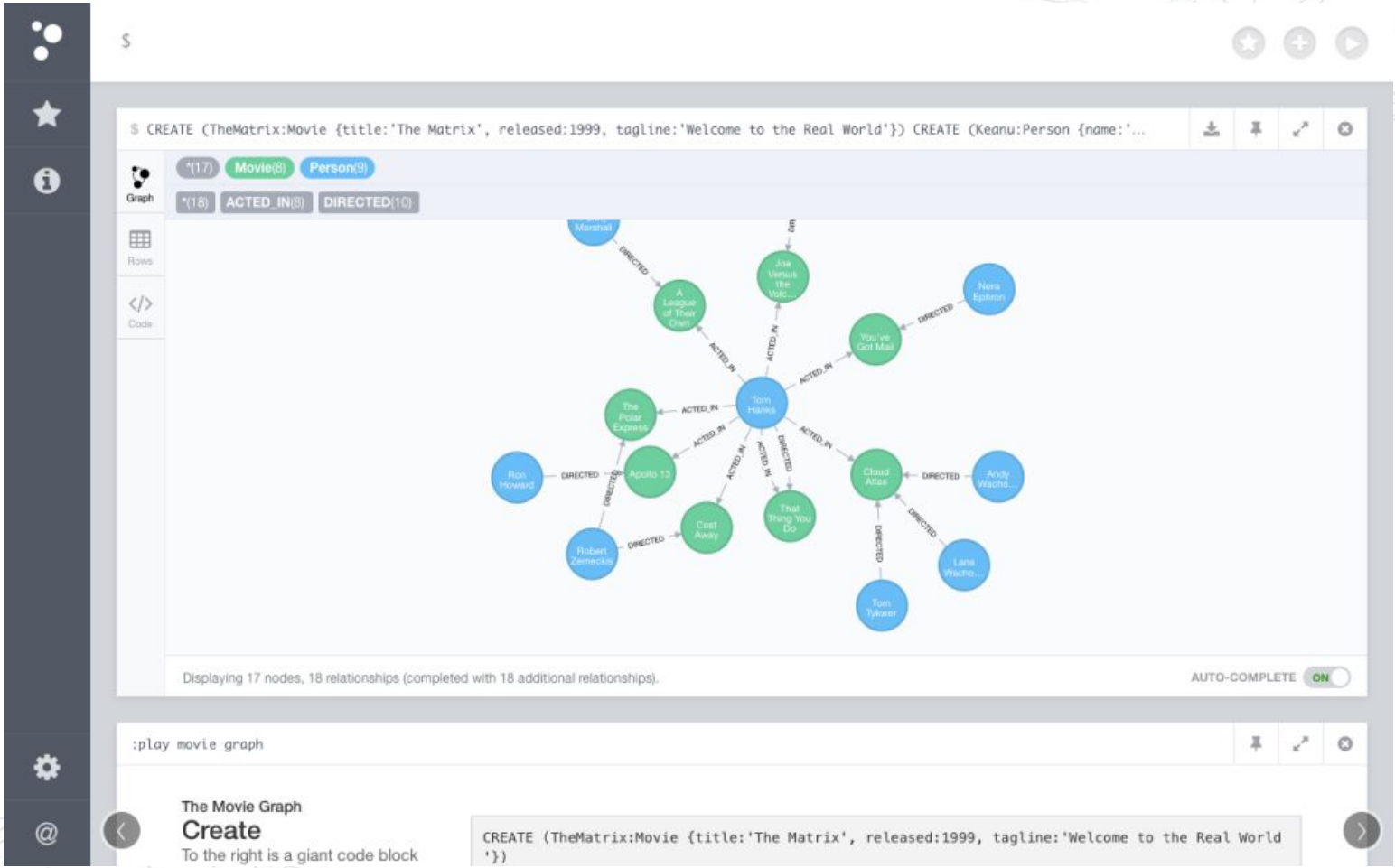
# Neo4J Graph Visualization

Leonid Scott, Shawn Saiyev

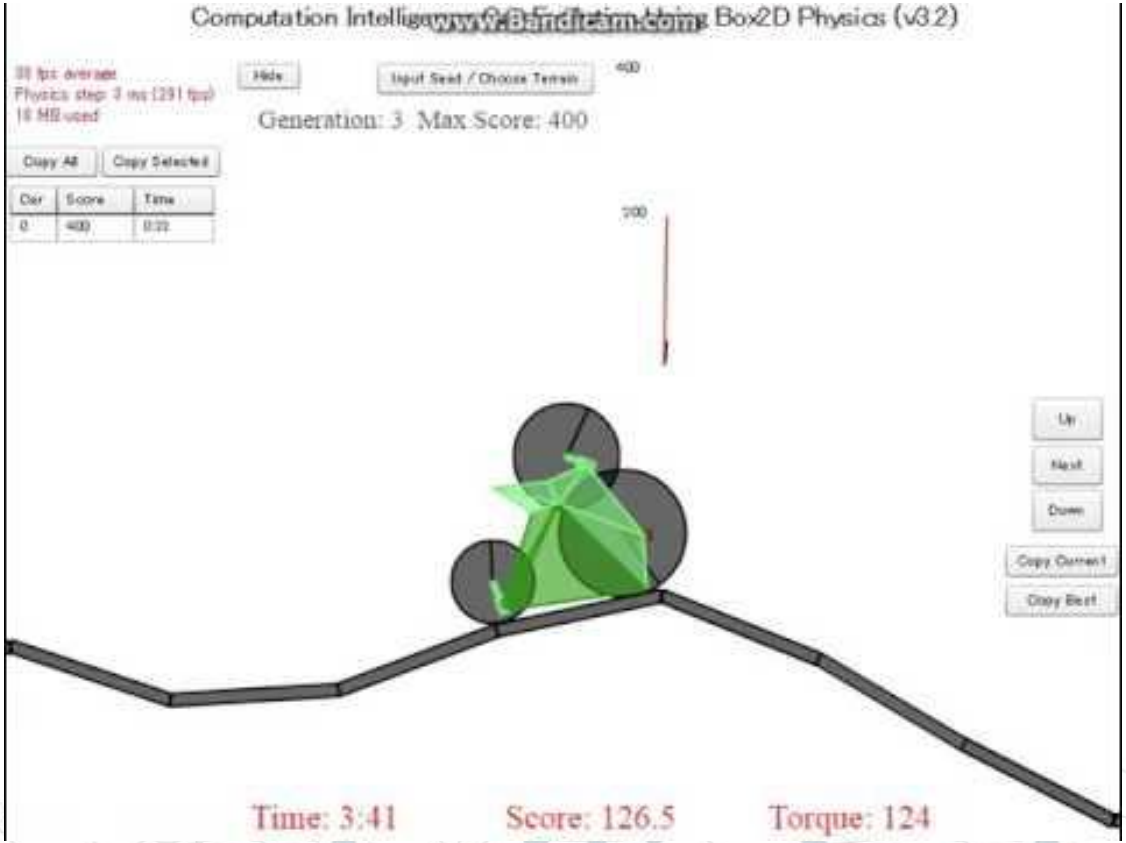
Databases Final Project F2017

# What is Neo4J?

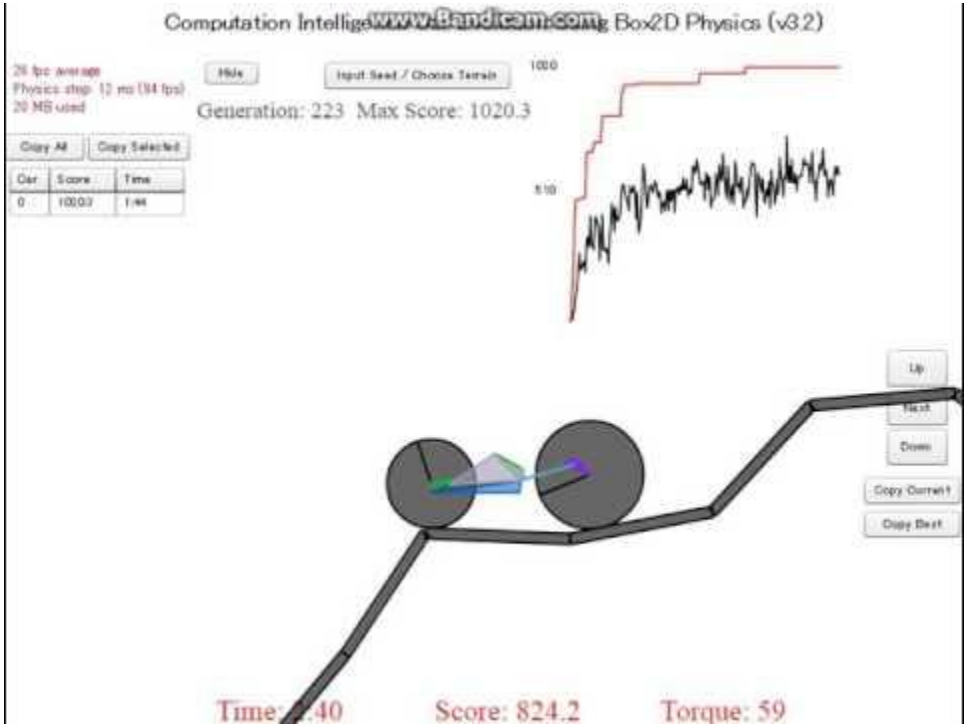

$
$ CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'}) CREATE (Keanu:Person {name:'...
Graph
Rows
Code
*(17)
Movie(8)
Person(9)
*(18)
ACTED_IN(8)
DIRECTED(10)
Marshall
A League of Their Own
Joe Versus the Volc...
Nora Ephron
You've Got Mail
Tom Hanks
The Polar Express
Ron Howard
Apollo 13
Cast Away
That Thing You Do
Cloud Atlas
Andy Wacho...
Robert Zemeckis
Lana Wacho...
Tom Tykwer
DIRECTED
ACTED_IN
Displaying 17 nodes, 18 relationships (completed with 18 additional relationships).
AUTO-COMPLETE
ON
:play movie graph
The Movie Graph
Create
To the right is a giant code block
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World
'})

# Motivation

# Evolutionary Computation (EC)

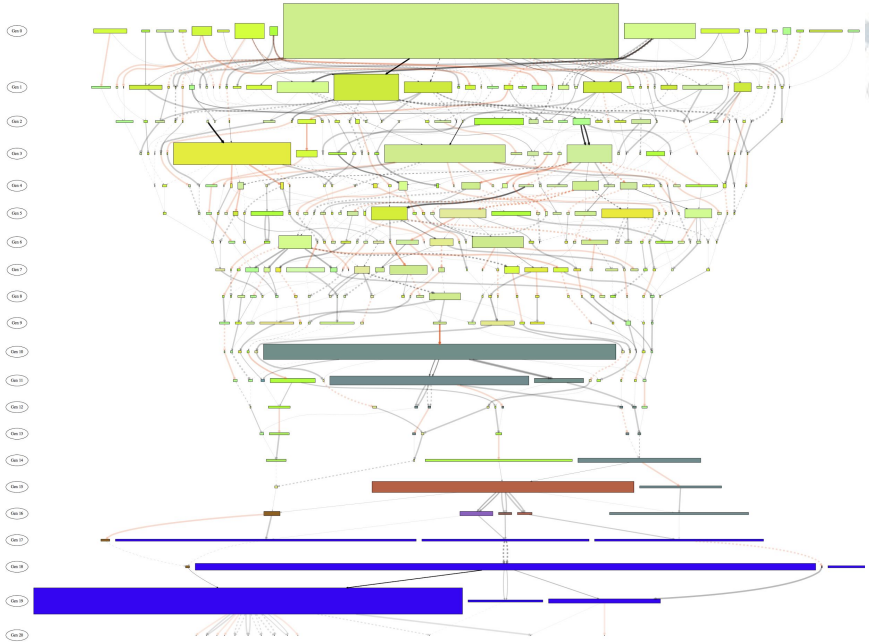


# Evolutionary Computation (EC)

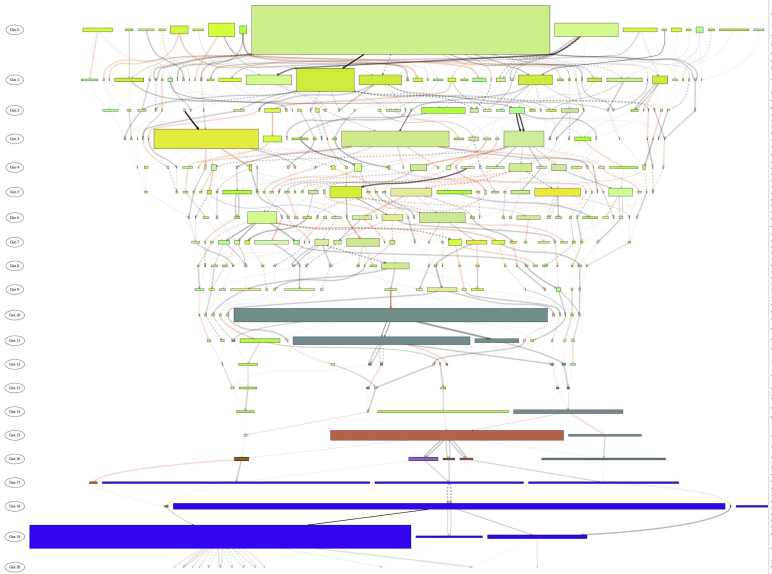


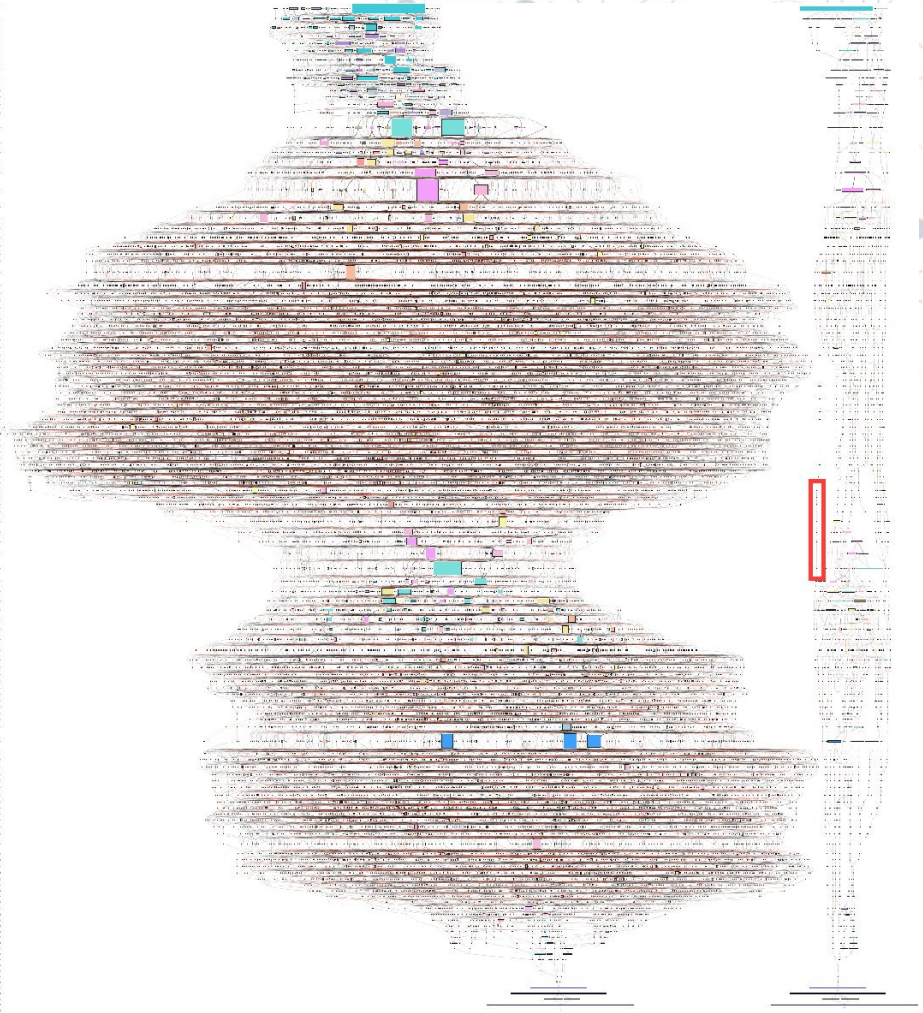
# McPhee, Specter Et.Al (Not necessarily in that order)

- Genetic Programming
- Nic McPhee adds these results into a graph database (currently neo4J)
- Nic McPhee visualizes these results in pdf’s.



# PDF Graphs





# Pro's and Con’s to PDF’s

Pros

- Good for a grand scale view of the run
- Because they are pdf’s you can zoom into subsections of the graph

Cons

- Slow to load, cumbersome to use
- Too packed full of info when not filtered
- Might miss info when filtered

# This is why we want to make an interactive graphing tool

- Use in conjunction with PDF’s
- View subsections in greater detail
- Create a good platform to share data

# Graphing Tools

# Graph Tools and Libraries

- Most difficult part of the project
- Graphing Tools use a *Force Directed Graph* (example)
- Tree making tools do not allow a child to have more than one parent. (example)

# D3 Implementations

Two implementations we considered:

- Cinima Force-Directed Graph:
  - Neo4J's D3 Example
  - Had some documentation

- DagreD3
  - Project made by cpettitt on github
  - Tree like structure, but with graph like properties
  - dagreD3 README " This project is not being actively developed or maintained"
- Settled on DagreD3, with Angular JS Framework, and express.js on the backend

# Implementation

# API

Our project may have got REST, be we didn’t.

We have two API requests

- /getWinners: Returns all individuals that solved the problem correctly
- /getAncestors?child_uuid: Returns all parents for a given individual.

# Neo4J Queries

Our project may have got REST, be we didn't.

We have two API requests

- /getWinners:

```
MATCH(n: Individual) - [HasTotalError] -> (x: TotalError {TotalError:  0 }) RETURN n LIMIT 100;
```

- /getAncestors?child_uuid:

```
MATCH (n: Individual)-[: ParentOf]->(i: Individual {uuid: \'"+ child_uuid +"\'}) RETURN n;
```

# Demo

Link (Here)