Lecture 30: Conclusion

Brian Hou
August 11, 2016

Announcements

Announcements

- Final Exam tomorrow (8/12) from 5–8pm in 155 Dwinelle

Announcements

- Final Exam tomorrow (8/12) from 5–8pm in 155 Dwinelle
- Last part of AutoStyle EC study is due today

Announcements

- Final Exam tomorrow (8/12) from 5–8pm in 155 Dwinelle
- Last part of AutoStyle EC study is due today
- Homework 12 out later today, due Saturday 8/13

Announcements

- Final Exam tomorrow (8/12) from 5–8pm in 155 Dwinelle
- Last part of AutoStyle EC study is due today
- Homework 12 out later today, due Saturday 8/13
  - End–of–semester survey, one more extra credit point!

Scheme Recursive Art Contest

http://art.cs61a.org/

Scheme Recursive Art Contest

Scheme Recursive Art Contest

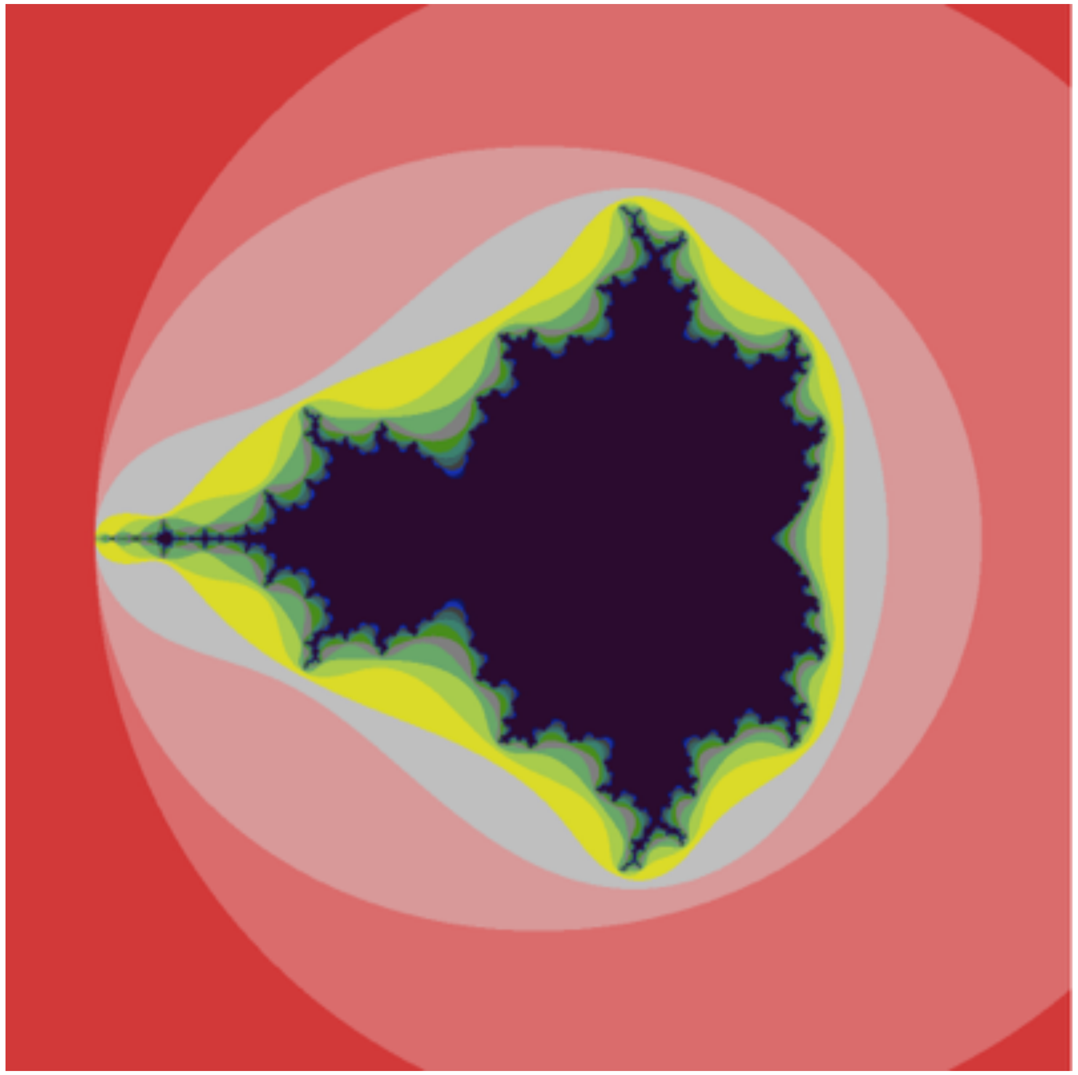
- Congratulations to everyone who participated in this semester's Scheme Recursive Art Contest!

Scheme Recursive Art Contest

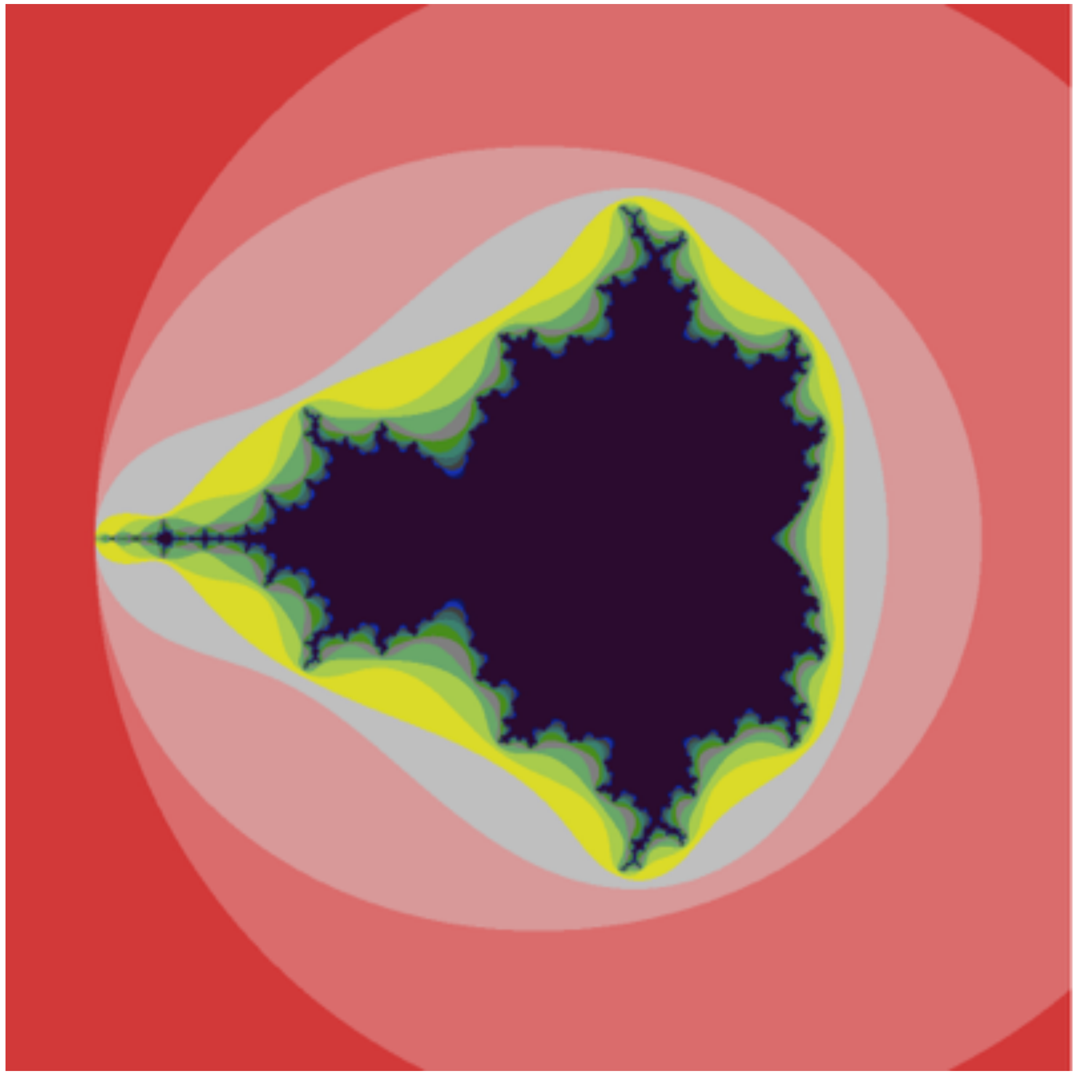
- Congratulations to everyone who participated in this semester's Scheme Recursive Art Contest!
- Thank you to everyone who helped us decide the winners!

Featherweight (Third Place)

Featherweight (Third Place)



Featherweight (Third Place)



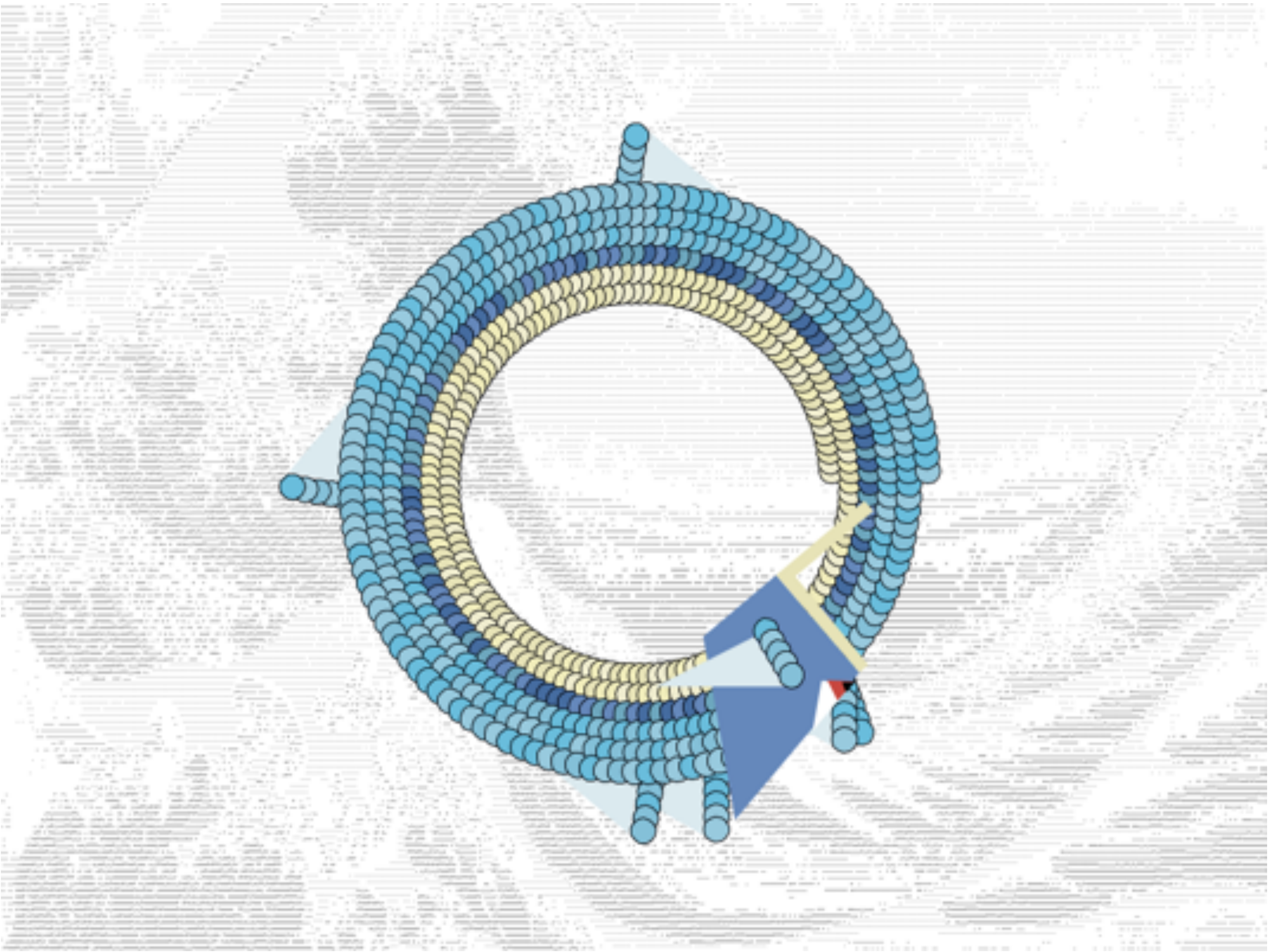
Mandelbrot Frrrrraction!!

Peilin Lu

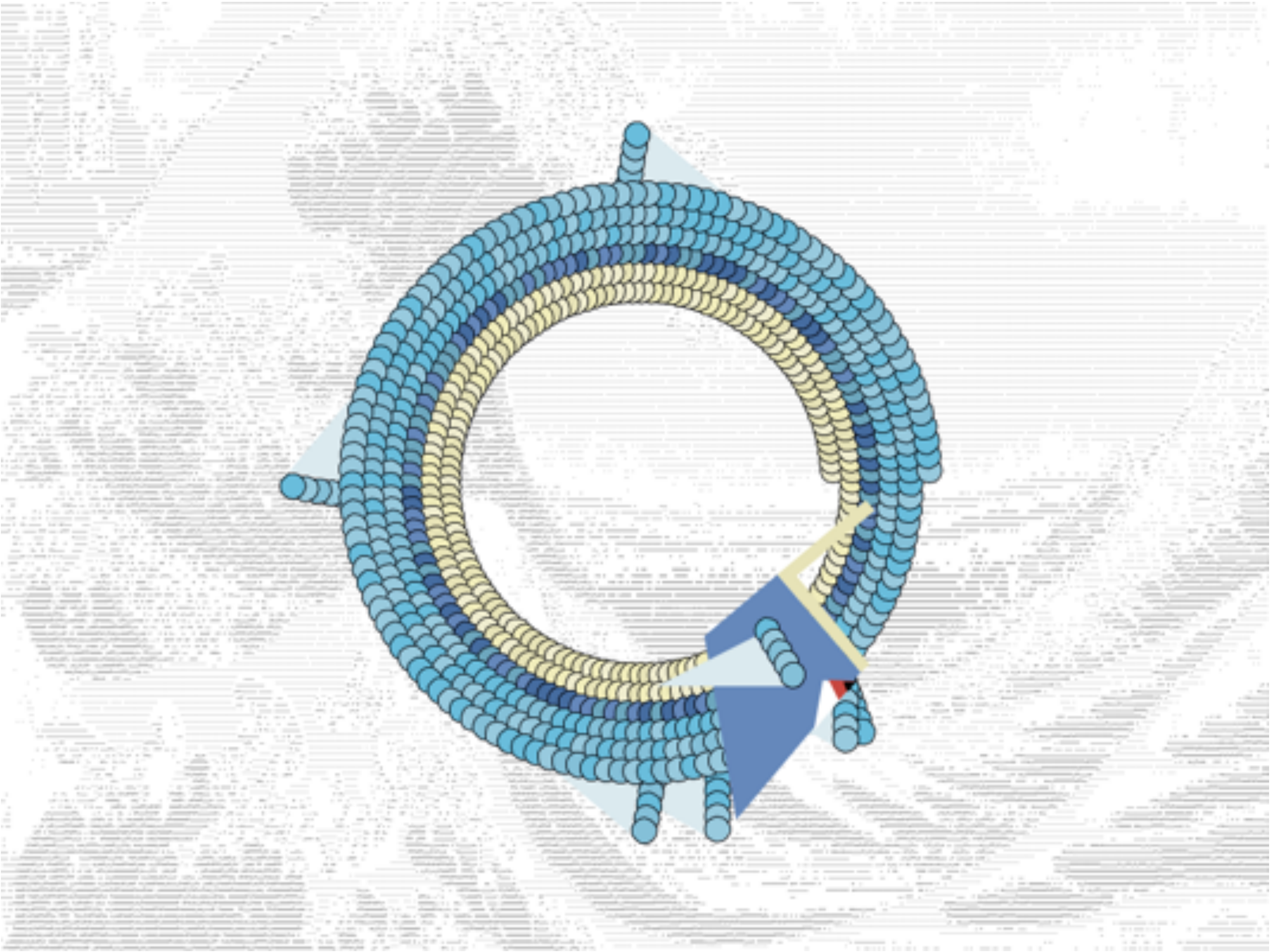
13.1% of votes

Featherweight (Second Place)

Featherweight (Second Place)



Featherweight (Second Place)



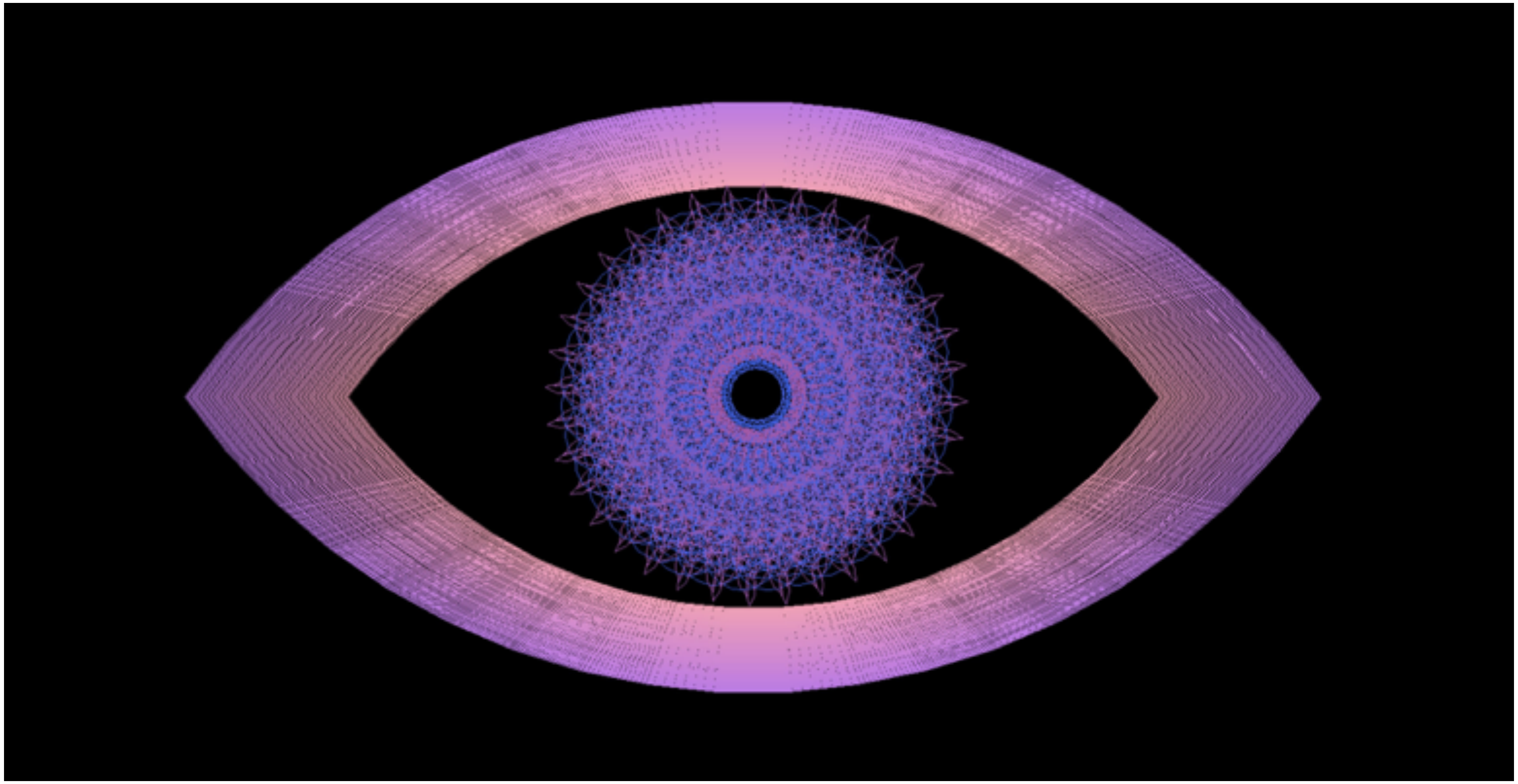
Tail–recursive Gyarados

Leo Adberg and Amir Shahatit

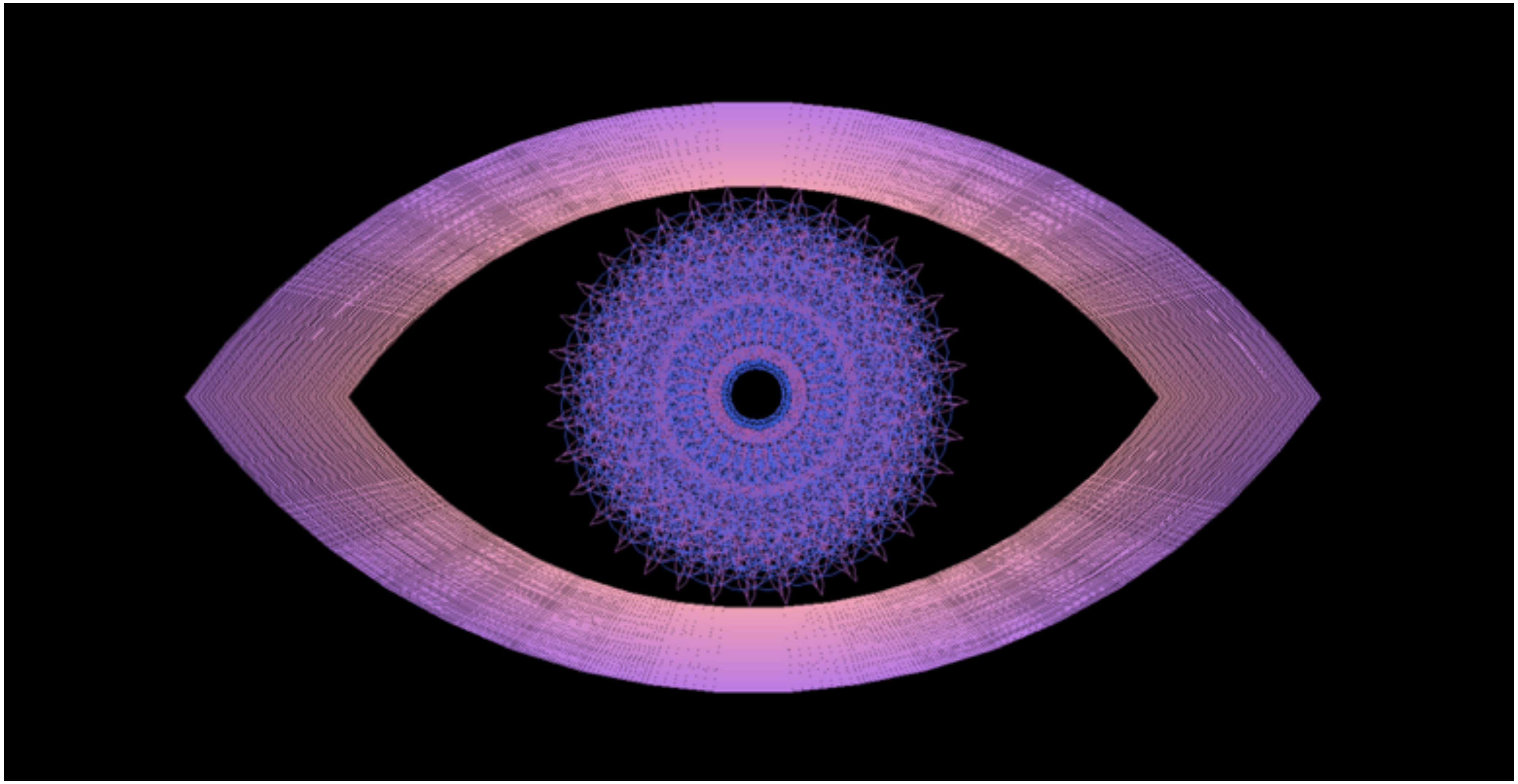
13.4% of votes

Featherweight (First Place)

Featherweight (First Place)



Featherweight (First Place)



Staring Eye

Renhua Liu

14.4% of votes

Heavyweight (Third Place)

Heavyweight (Third Place)

Heavyweight (Third Place)

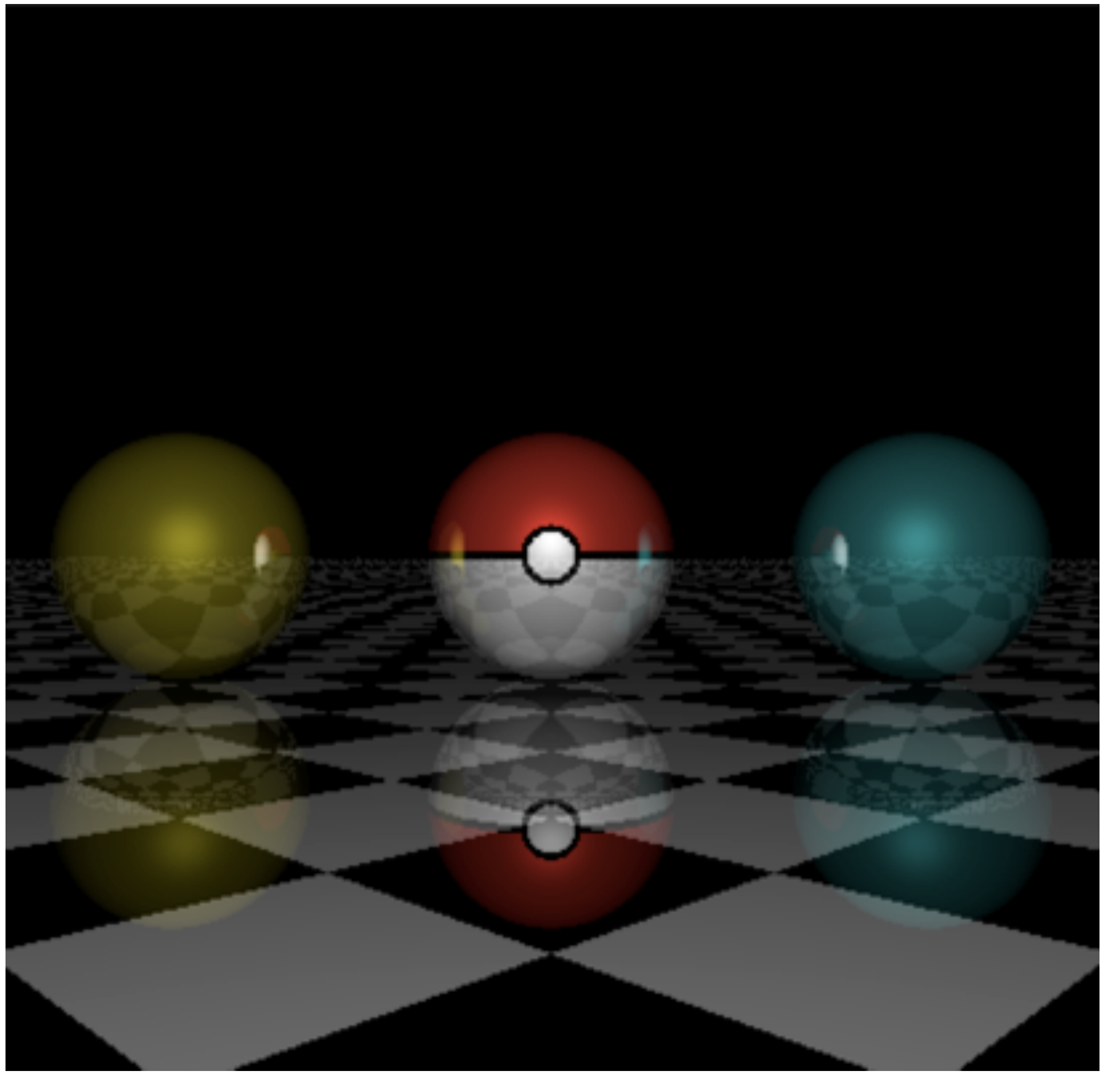
Vigil for The Person Who Got –5 Points in CS61A

Xiaocheng Yang and Zeyana Musthafa

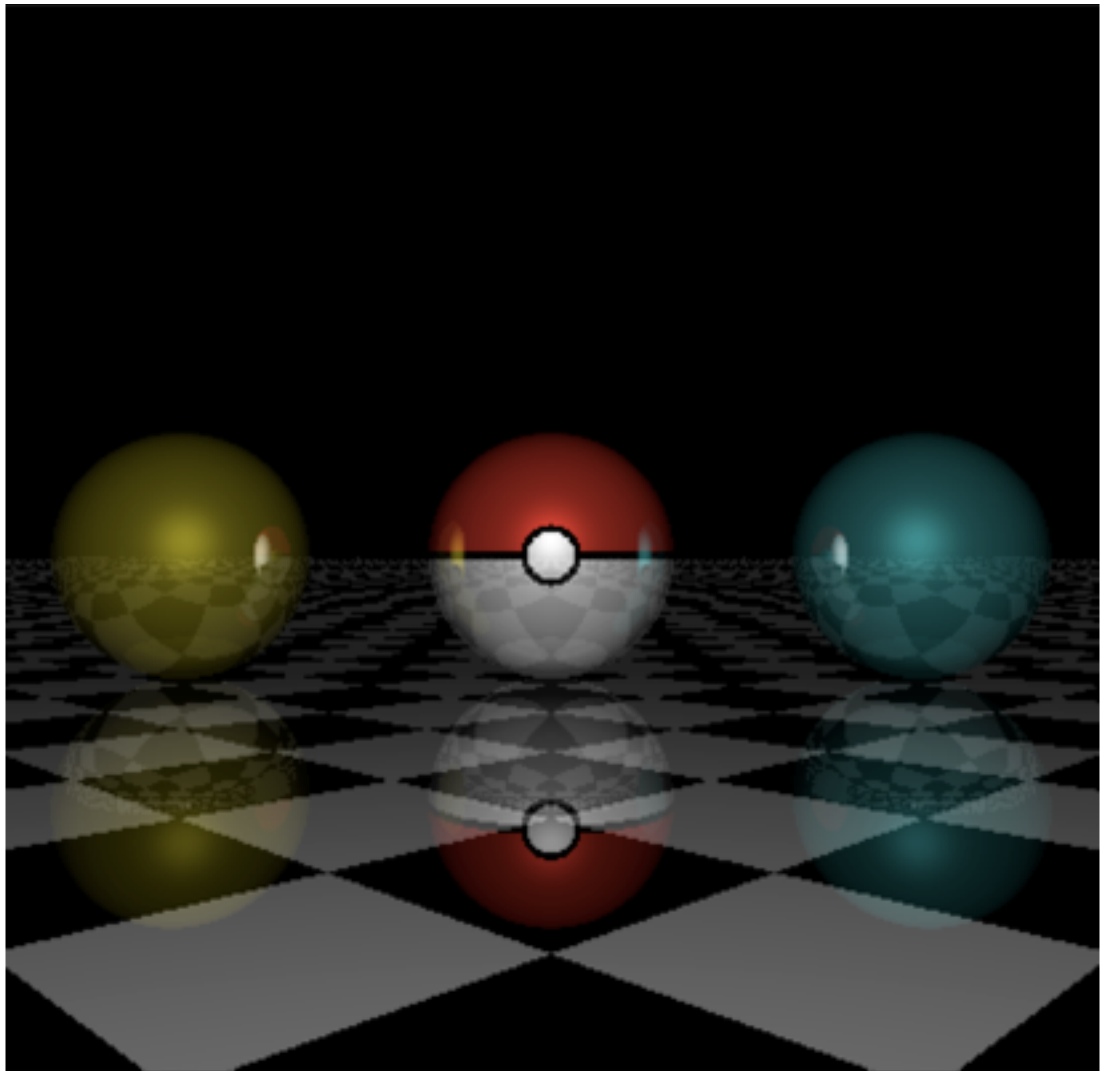
14.1% of votes

Heavyweight (Second Place)

Heavyweight (Second Place)



Heavyweight (Second Place)



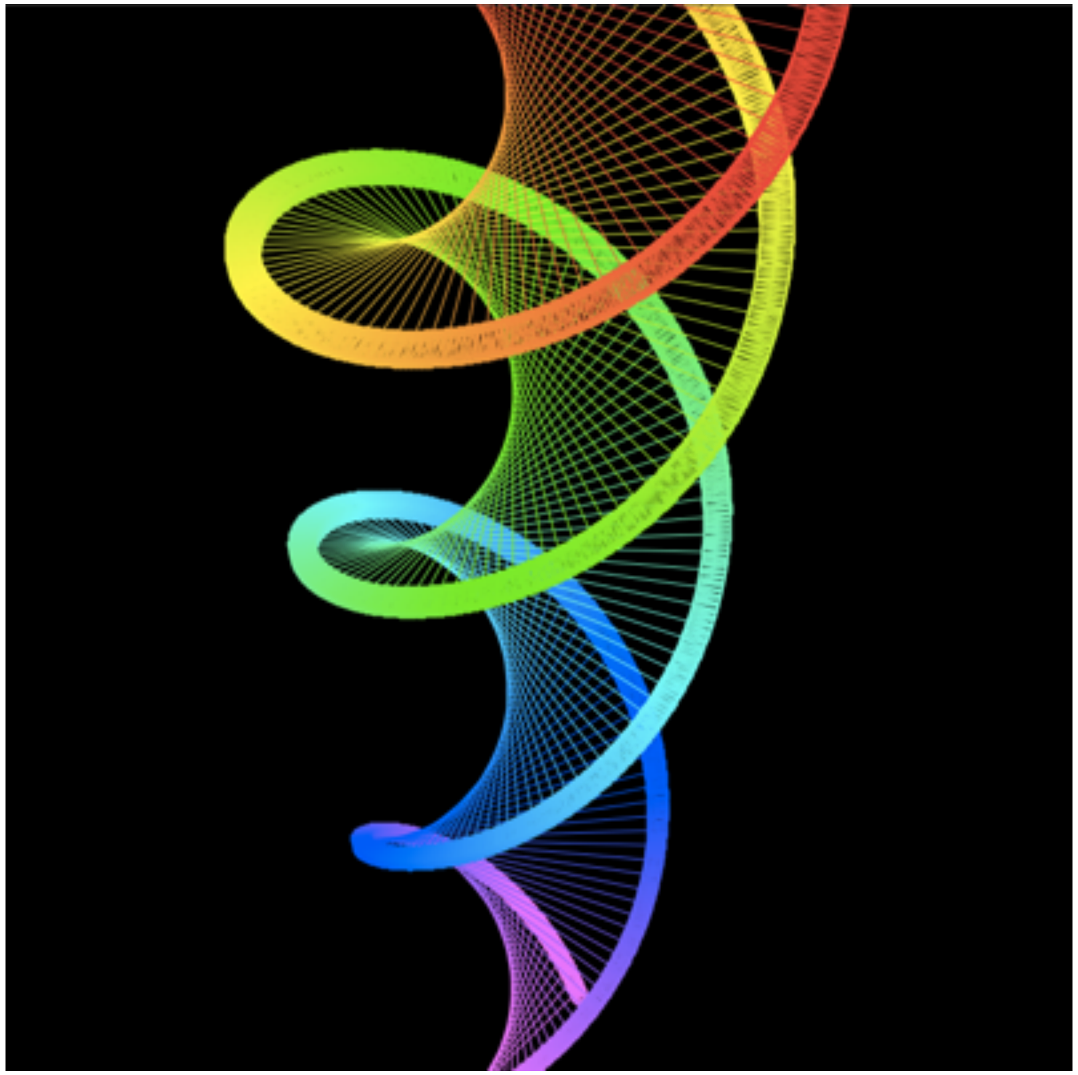
EE/CS Master Trainers

Alex Bondarenko

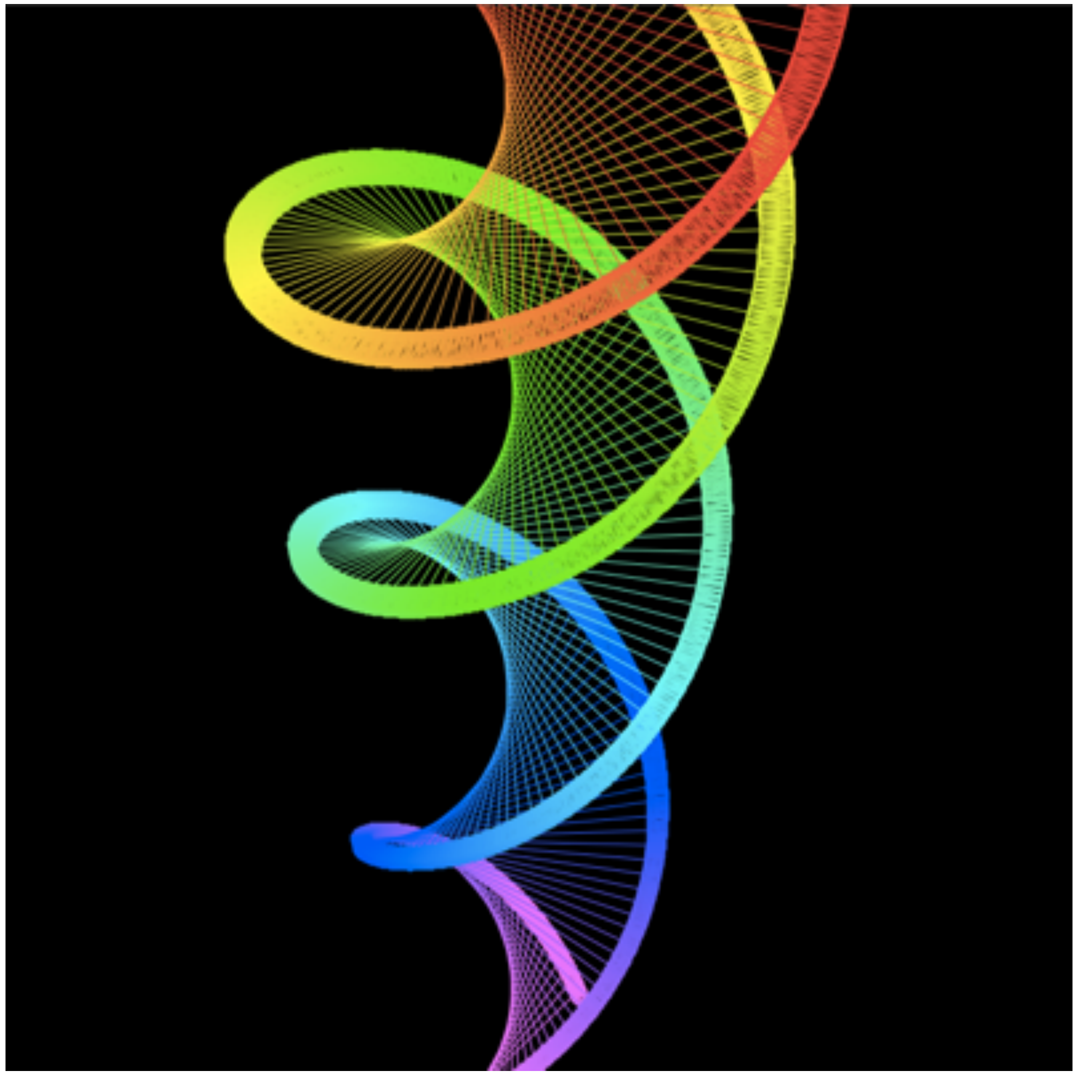
28.4% of votes

Heavyweight (First Place)

Heavyweight (First Place)



Heavyweight (First Place)

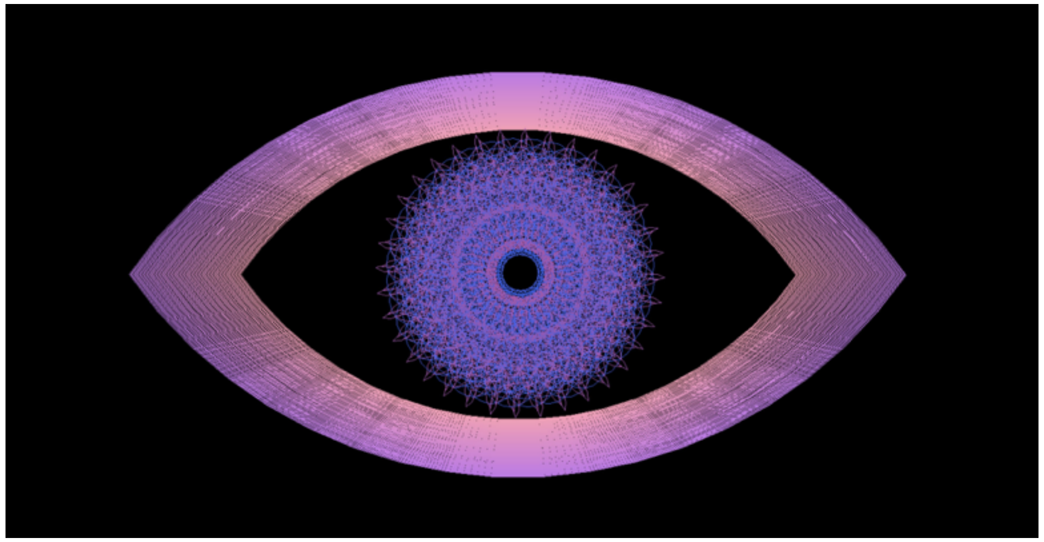


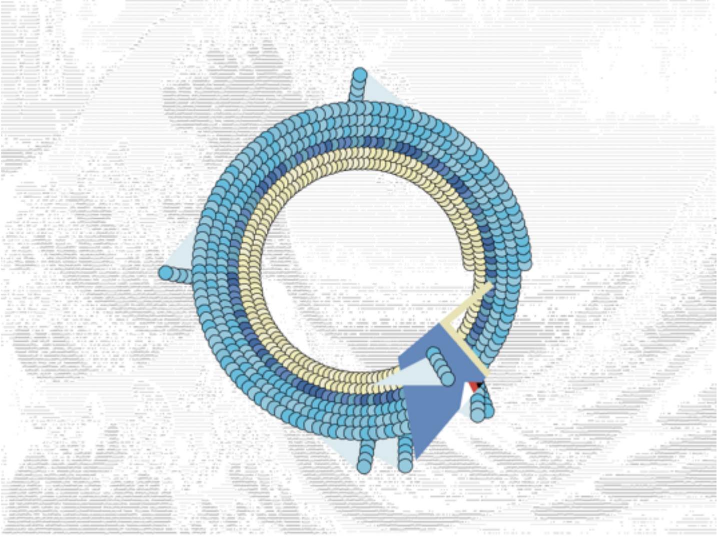
Origin of Life

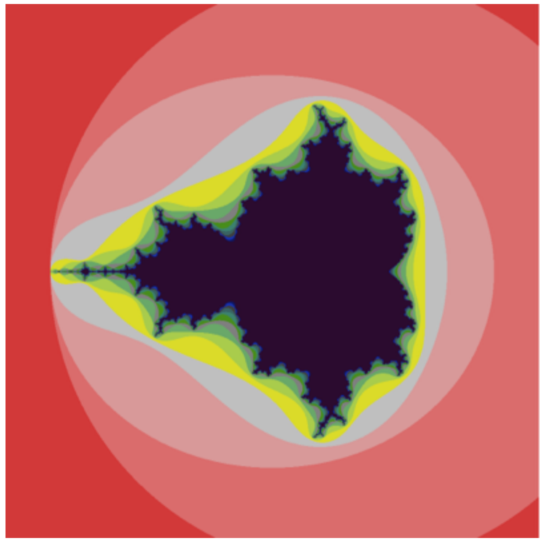
Yi Xu and Jianhui Li

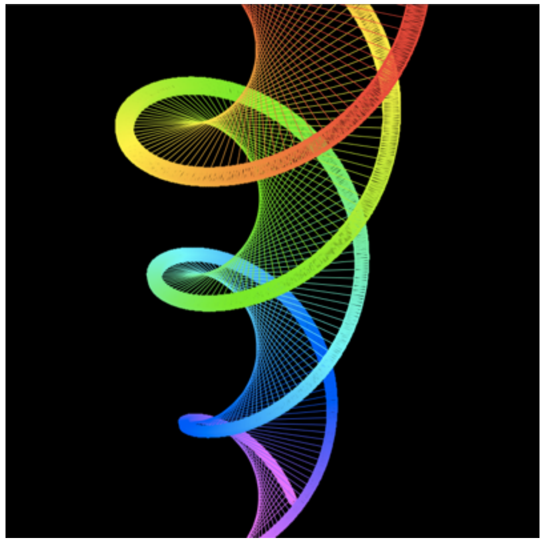
30.0% of votes

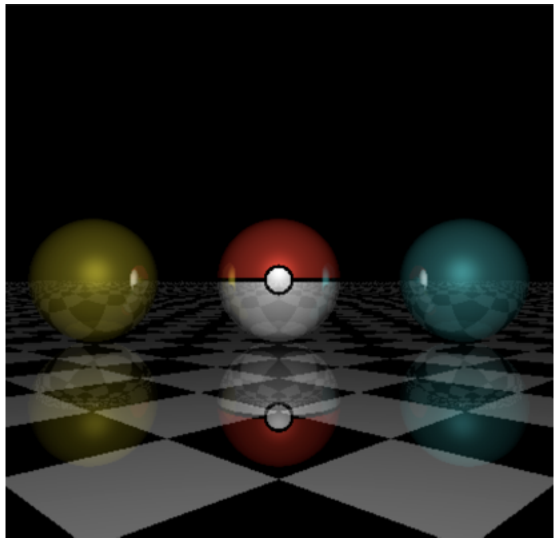
Congratulations!











What is CS 61A?

CS 61A in one slide

CS 61A in one slide

- High–level ideas in computer science:

CS 61A in one slide

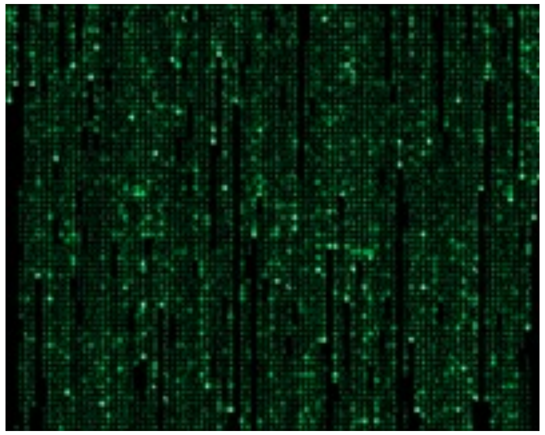
- High–level ideas in computer science:
  - *Abstraction*: manage complexity by hiding the details

CS 61A in one slide

- High–level ideas in computer science:
  - *Abstraction*: manage complexity by hiding the details
  - *Paradigms*: utilize different approaches to programming

CS 61A in one slide

- High-level ideas in computer science:
  - *Abstraction*: manage complexity by hiding the details
  - *Paradigms*: utilize different approaches to programming



CS 61A in one slide

- High–level ideas in computer science:
  - *Abstraction*: manage complexity by hiding the details
  - *Paradigms*: utilize different approaches to programming

CS 61A in one slide

- High–level ideas in computer science:
  - *Abstraction*: manage complexity by hiding the details
  - *Paradigms*: utilize different approaches to programming

- Master these ideas through implementation:

CS 61A in one slide

- High–level ideas in computer science:
  - *Abstraction*: manage complexity by hiding the details
  - *Paradigms*: utilize different approaches to programming
- Master these ideas through implementation:
  - Learn the Python programming language (& others)

CS 61A in one slide

- High–level ideas in computer science:
    - *Abstraction*: manage complexity by hiding the details
    - *Paradigms*: utilize different approaches to programming

- Master these ideas through implementation:
    - Learn the Python programming language (& others)
    - Complete large programming assignments

CS 61A in one slide

- High–level ideas in computer science:
  - *Abstraction*: manage complexity by hiding the details
  - *Paradigms*: utilize different approaches to programming

- Master these ideas through implementation:
  - Learn the Python programming language (& others)
  - Complete large programming assignments

- A challenging course that will demand a lot from you

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Introduction), the goals are:

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Introduction), the goals are:
  - To learn the fundamentals of programming

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Introduction), the goals are:
  - To learn the fundamentals of programming
  - To become comfortable with Python

Roadmap

- Introduction
- Functions
- Data
- Mutability
- Objects
- Interpretation
- Paradigms
- Applications

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Functions), the goals are:

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Functions), the goals are:
  - To understand the idea of *functional abstraction*

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Functions), the goals are:
  - To understand the idea of *functional abstraction*
  - To study this idea through:

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Functions), the goals are:
  - To understand the idea of *functional abstraction*
  - To study this idea through:
    - higher-order functions

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Functions), the goals are:
  - To understand the idea of *functional abstraction*
  - To study this idea through:
    - higher-order functions
    - recursion

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Functions), the goals are:
  - To understand the idea of *functional abstraction*
  - To study this idea through:
    - higher-order functions
    - recursion
    - orders of growth

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Data), the goals are:

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Data), the goals are:
  - To continue our journey through abstraction with *data abstraction*

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Data), the goals are:
  - To continue our journey through abstraction with *data abstraction*
  - To study useful data types we can construct with data abstraction

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This short week (Mutability), the goals are:

Roadmap

- Introduction
- Functions
- Data
- Mutability
- Objects
- Interpretation
- Paradigms
- Applications

- This short week (Mutability), the goals are:
    - To explore the power of values that can *mutate*, or change

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Objects), the goals are:

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Objects), the goals are:
  - To learn the paradigm of *object-oriented programming*

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Objects), the goals are:
  - To learn the paradigm of *object-oriented programming*
  - To study applications of, and problems that be solved using, OOP

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

Roadmap

- Introduction
- Functions
- Data
- Mutability
- Objects
- Interpretation
- Paradigms
- Applications

- This week (Interpretation), the goals are:

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Interpretation), the goals are:
    - To learn a new language, Scheme, in two days!

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Interpretation), the goals are:
  - To learn a new language, Scheme, in two days!
  - To understand how interpreters work, using Scheme as an example

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

Roadmap

- Introduction
- Functions
- Data
- Mutability
- Objects
- Interpretation
- Paradigms
- Applications

- This week (Paradigms), the goals are:

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Paradigms), the goals are:
  - To study examples of paradigms that are very different from what we have seen so far

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Paradigms), the goals are:
  - To study examples of paradigms that are very different from what we have seen so far
  - To expand our definition of what counts as programming

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Applications), the goals are:

Roadmap

- Introduction
- Functions
- Data
- Mutability
- Objects
- Interpretation
- Paradigms
- Applications

- This week (Applications), the goals are:
  - To go beyond CS 61A and see examples of what comes next

Roadmap

Introduction

Functions

Data

Mutability

Objects

Interpretation

Paradigms

Applications

- This week (Applications), the goals are:
  - To go beyond CS 61A and see examples of what comes next
  - To wrap up CS 61A!

Life After CS 61A

Classes at Berkeley

Classes at Berkeley

- What you learn is much more important than your grade!

Classes at Berkeley

- What you learn is much more important than your grade!
- CS 61B (Data Structures and Algorithms)

Classes at Berkeley

- What you learn is much more important than your grade!
- CS 61B (Data Structures and Algorithms)
  - Taught by Professor Paul Hilfinger in Fall 2016

Classes at Berkeley

- What you learn is much more important than your grade!
- CS 61B (Data Structures and Algorithms)
  - Taught by Professor Paul Hilfinger in Fall 2016
- Data Science 8 (Foundations of Data Science)

Classes at Berkeley

- What you learn is much more important than your grade!
- CS 61B (Data Structures and Algorithms)
  - Taught by Professor Paul Hilfinger in Fall 2016
- Data Science 8 (Foundations of Data Science)
  - Taught by Professor Ani Adhikari in Fall 2016

Classes at Berkeley

- What you learn is much more important than your grade!
- CS 61B (Data Structures and Algorithms)
  - Taught by Professor Paul Hilfinger in Fall 2016
- Data Science 8 (Foundations of Data Science)
  - Taught by Professor Ani Adhikari in Fall 2016
- Other EECS lower division courses:

Classes at Berkeley

- What you learn is much more important than your grade!
- CS 61B (Data Structures and Algorithms)
  - Taught by Professor Paul Hilfinger in Fall 2016
- Data Science 8 (Foundations of Data Science)
  - Taught by Professor Ani Adhikari in Fall 2016
- Other EECS lower division courses:
  - CS 70 (Discrete Mathematics and Probability Theory)

Classes at Berkeley

- What you learn is much more important than your grade!
- CS 61B (Data Structures and Algorithms)
  - Taught by Professor Paul Hilfinger in Fall 2016
- Data Science 8 (Foundations of Data Science)
  - Taught by Professor Ani Adhikari in Fall 2016
- Other EECS lower division courses:
  - CS 70 (Discrete Mathematics and Probability Theory)
  - CS 61C (Machine Structures)

Classes at Berkeley

- What you learn is much more important than your grade!
- CS 61B (Data Structures and Algorithms)
  - Taught by Professor Paul Hilfinger in Fall 2016
- Data Science 8 (Foundations of Data Science)
  - Taught by Professor Ani Adhikari in Fall 2016
- Other EECS lower division courses:
  - CS 70 (Discrete Mathematics and Probability Theory)
  - CS 61C (Machine Structures)
  - EE 16A/16B (Designing Information Devices and Systems)

Classes at Berkeley

- What you learn is much more important than your grade!
- CS 61B (Data Structures and Algorithms)
  - Taught by Professor Paul Hilfinger in Fall 2016
- Data Science 8 (Foundations of Data Science)
  - Taught by Professor Ani Adhikari in Fall 2016
- Other EECS lower division courses:
  - CS 70 (Discrete Mathematics and Probability Theory)
  - CS 61C (Machine Structures)
  - EE 16A/16B (Designing Information Devices and Systems)
- EECS upper division courses

Life Outside the Classroom

Life Outside the Classroom

- Program for fun! Build things that *you* think are cool

Life Outside the Classroom

- Program for fun! Build things that *you* think are cool
  - Hackathons are a great place for this to happen

Life Outside the Classroom

- Program for fun! Build things that *you* think are cool
  - Hackathons are a great place for this to happen
- Try an internship or join a research project

Life Outside the Classroom

- Program for fun! Build things that *you* think are cool
  - Hackathons are a great place for this to happen
- Try an internship or join a research project
- Don't forget to do things that aren't CS–related!

Lab Assisting

Lab Assisting

- The best way to give back to the CS community

Lab Assisting

- The best way to give back to the CS community
- Anyone who passes the course can be a lab assistant

Lab Assisting

- The best way to give back to the CS community
- Anyone who passes the course can be a lab assistant
- Develop greater mastery of course concepts

Lab Assisting

- The best way to give back to the CS community
- Anyone who passes the course can be a lab assistant
- Develop greater mastery of course concepts
- Learn to describe technical concepts (great preparation for technical interviews!)

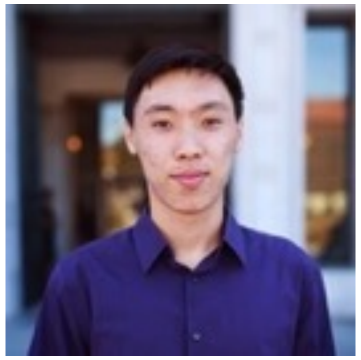
Lab Assisting

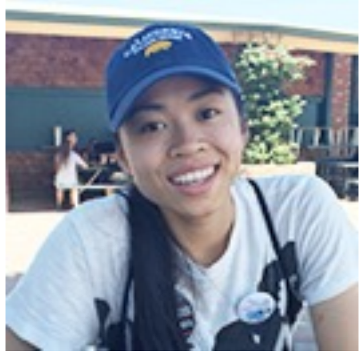
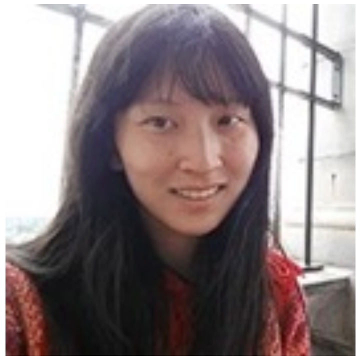
- The best way to give back to the CS community
- Anyone who passes the course can be a lab assistant
- Develop greater mastery of course concepts
- Learn to describe technical concepts (great preparation for technical interviews!)
- The first step to joining the course staff as a tutor or teaching assistant

Lab Assisting

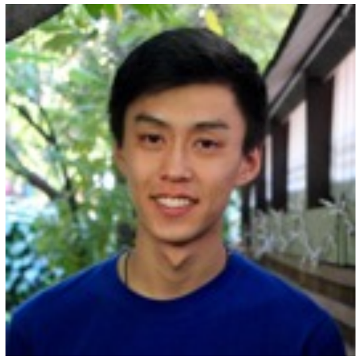
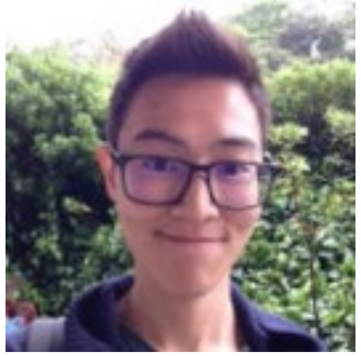
- The best way to give back to the CS community
- Anyone who passes the course can be a lab assistant
- Develop greater mastery of course concepts
- Learn to describe technical concepts (great preparation for technical interviews!)
- The first step to joining the course staff as a tutor or teaching assistant

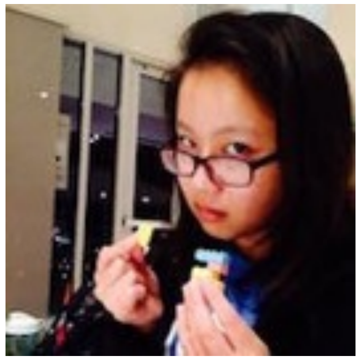
https://piazza.com/class/ipkfex1ne3p56y?cid=1682

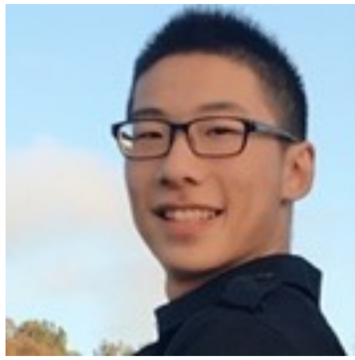


Thank you!





Q & A