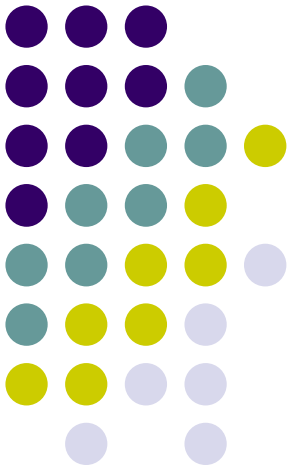
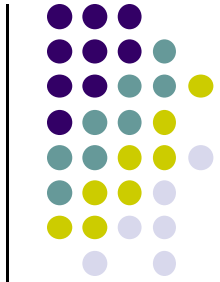
CS 403X Mobile and Ubiquitous Computing

Lecture 13: Presentation/Summary Guidelines

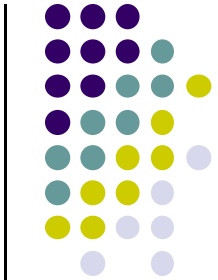
Emmanuel Agu





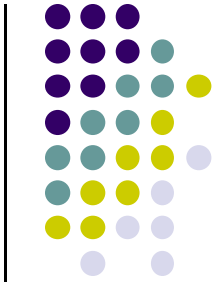
Presentation Guidelines

Overview



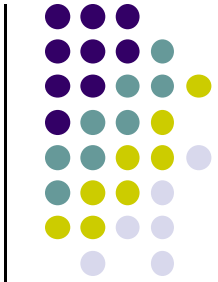
- Next week, class enters new phase of class featuring
  - Paper presentations (starting April 11)
  - Writing critiques of papers
  - Final projects
- Each class, 2 presentations in groups (see presentation schedule)
- **Students should summarize/critique any 4 papers** excluding the paper that they present.
- Next, I provide guidelines on presenting papers and writing critiques

Your Presentation



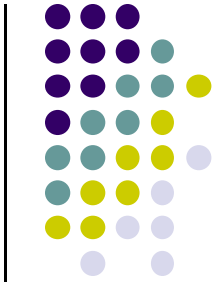
- About 20 mins
- Estimate: about 2 mins per slide
- About 10 slides should be enough + **front page and references** (~ 12 pages)
- Allow 5 mins for questions, discussions

Main Points Presentation Should Cover



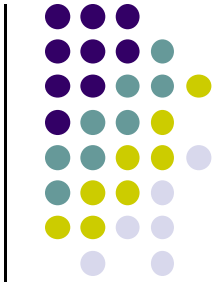
- Introduction/motivation:
  - What was the main problem addressed by the paper?
  - Why is the problem solved important
  - How will the solution be used eventually? How will this new approach save time, resources, incovenience, etc?
  - Focus on what is **new**:
    - scientific results, what was learned
    - Engineering results: new design + justification for choices

Main Points Presentation Should Cover



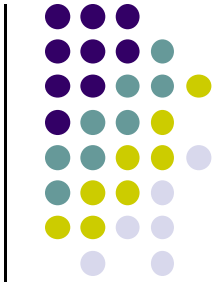
- Related Work:
  - What have other researchers done to solve this problem?
  - How is the approach proposed in this paper different or novel?
    - New idea:
    - New algorithm
    - New technique
    - New experiments

Main Points Presentation Should Cover



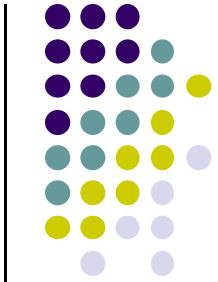
- Methodology/Approach:
  - Summarize the approach/design
  - If a system is described **“how does it work?”**
  - Describe the implementation used (languages, libraries, etc)
  - State any assumptions of the authors and limitations of the proposed work
  - What are the design tradeoffs?

Main Points Presentation Should Cover

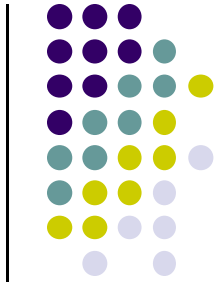


- Results:
  - Present a few of the most significant results/graphs
  - Results should show how well proposed approach worked or findings
  - Do the presented results back up the claims of the authors?

Main Points Presentation Should Cover

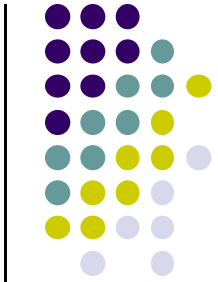


- Discussions/Conclusions/Future Work
  - Summarize what was achieved
  - What did you learn from this paper?
  - What extensions do the authors plan for future work?
  - Brief comments on the paper



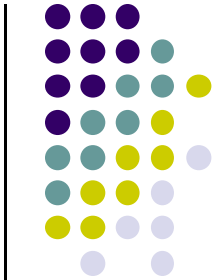
Paper Summary Guidelines

Paper Summary Guidelines

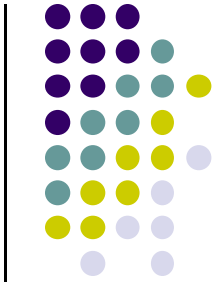


- Capture key points of paper
- Should not exceed half a page
- Don’t just cut-and-paste abstract blindly
- In 1 year’s time, summary should recall key aspects of paper, refresh memory without re-reading paper
- Provide key important details:
  - New idea, concepts, algorithms tools proposed?
- See guidelines on course website

Paper Summary Guidelines (Contd)



- Are assumptions fine?
- Design trade-offs?
- How is the organization of the paper, clarity of writing?
- Did the graphs, results support the claims by authors?
- What was good/Bad about paper?
- Suggestions for improvement?



Final Project Overview & Proposal Guidelines

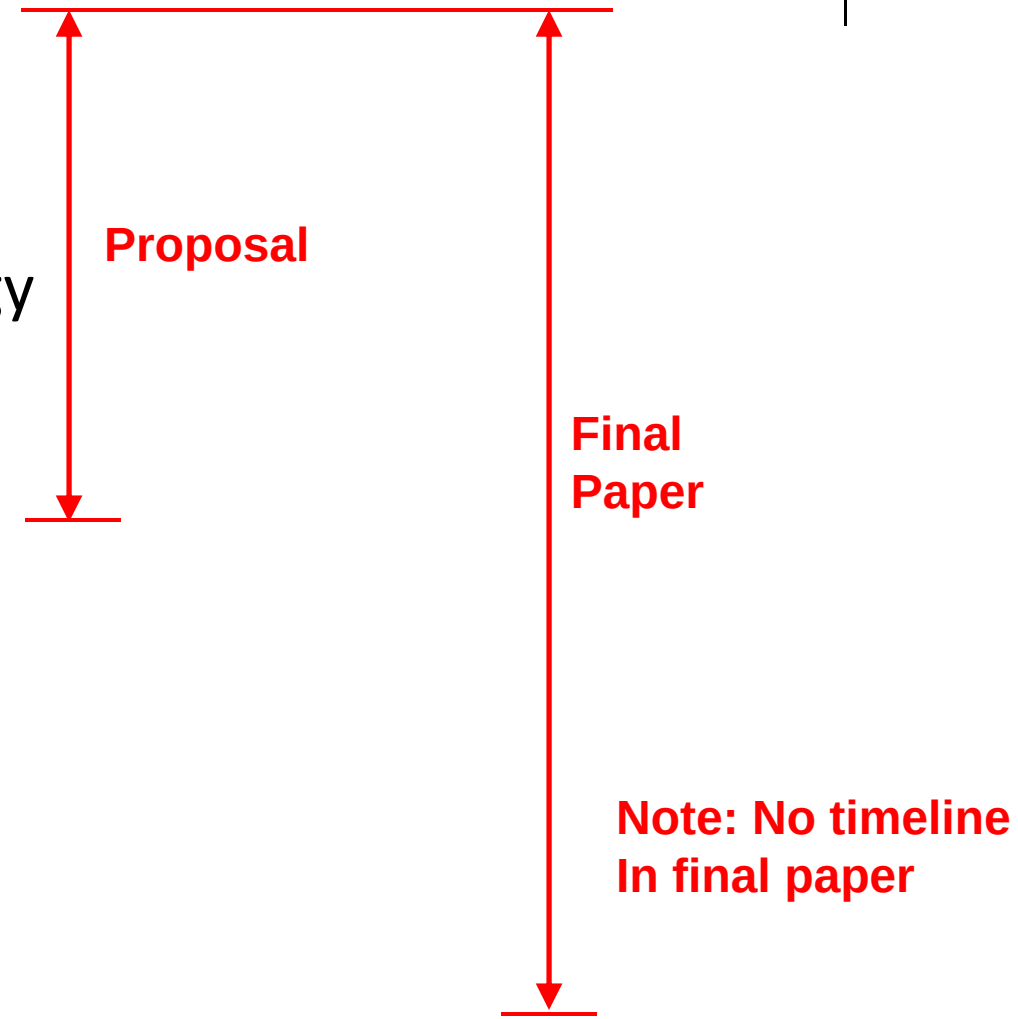
Final Project

- Most projects will probably build an app
- App solves some societal problem
- App should be **mobile** or/and **ubicomp**
  - **Mobile?** location-dependent, maps, deliver time-sensitive information
  - **Ubicomp?** Uses at least 1 sensor (accelerometer, microphone, camera, etc)
- Don't build app that has no mobile or ubicomp aspects
- If you have questions, talk to me

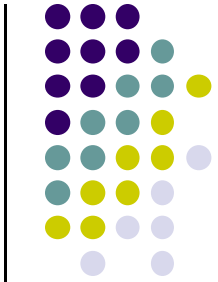
Typical Paper

- Introduction
- Related Work
- Approach/methodology
- Implementation
- **Project timeline**

- Evaluation/Results
- Discussion
- Conclusion
- Future Work

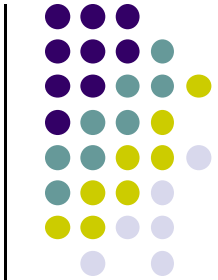


Proposal



- Submit (Written 2 pages max PDF file): due Apr 14!!
  - Introduction
    - List team members
    - State problem app will solve. Preferably has social benefit
    - Why is problem important?
    - E.g. Find statistics: How much time, money, resources is being wasted on this problem today? How many people problem affects
    - Potential gain: how will your solution save time, money, etc?
  - Related work
    - What other research has been done to solve this problem (academic + commercial apps)
    - How is your app/approach/work different?

Proposal

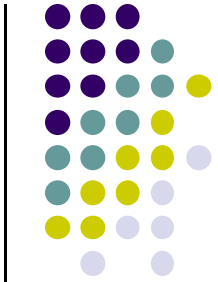


- Methodology/Design/Tools:
  - Brain storm!
  - Summary of what you intend to do
  - How you intend to do it? Build android app, use scenario, etc
  - App screen mock-ups:
    - Hand-drawn? Android Studio? Lucid Charts?
  - Don’t promise too much,
    - Some features can be future work

Methodology

- Preliminary design from team
- Screen mock-ups + flow
- Use Android Studio Design view, lucidcharts.com, hand-drawn?

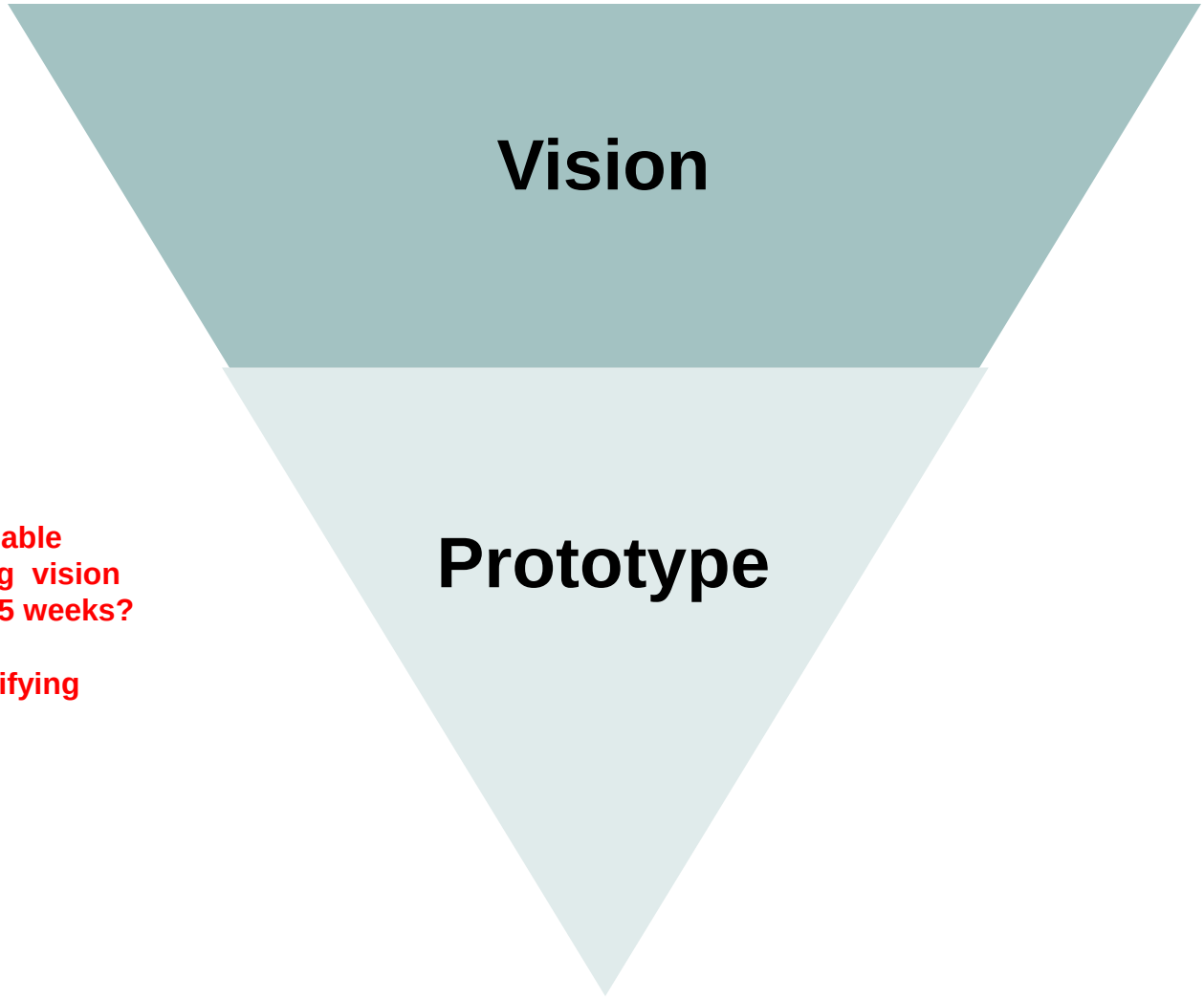
Proposal



- Implementation **plan**:
  - E.g. Android, what modules? external tools? Packages? etc
- Timeline
  - Break down tasks, mini-deadlines, allot time for each task
- Proposal due April 14!!

Separate Vision and Prototype

1. Big picture if funds/time not an issue (e.g. company of 200 employees over 6 years)



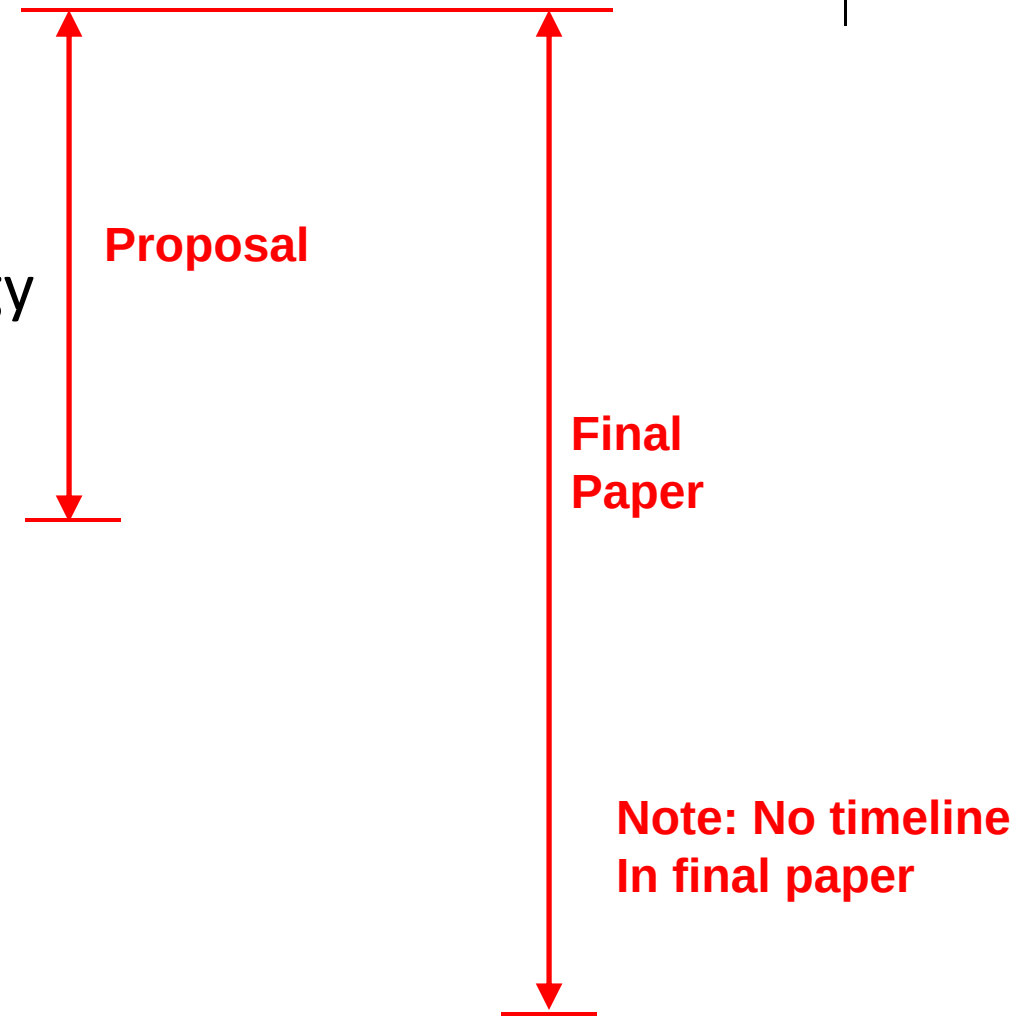
2. Which reasonable Subset of the big vision can you do in 2.5 weeks?

Can make simplifying assumptions

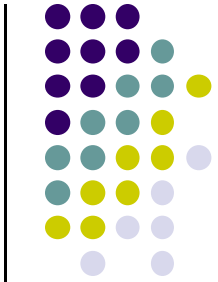
Typical Paper

- Introduction
- Related Work
- Approach/methodology
- Implementation
- **Project timeline**

- Evaluation/Results
- Discussion
- Conclusion
- Future Work



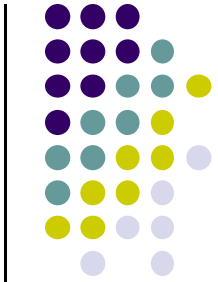
Final Paper: Evaluation



- Depends on what your project is.
- **Basic question:** How well did your solution work?
  - User studies
  - Measure performance. E.g. energy consumption, bandwidth consumption, etc

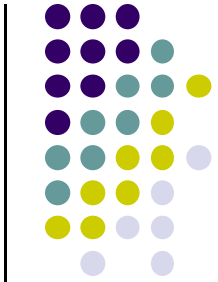
- User Studies
  - Pre-Survey:
    - Establish problem exists, need for your app, gather/refine requirements
  - Post-Survey:
    - Get users to use/rate your app, ask about likes dislikes

Recruiting Subjects For User Studies



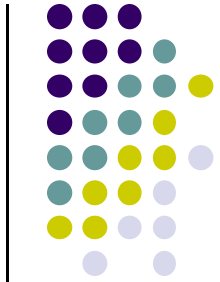
- 3Fs: Friends, Family and ??
- Classmates (Do a trade with another group)
- On campus: post flyers, set up table at campus center

Discussion, Conclusion, Future Work



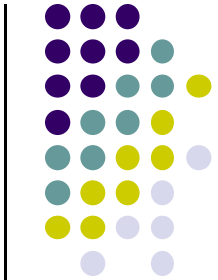
- Discussion:
  - How was your app received? Rationalize your findings in user studies, Say why certain features worked, did not work, etc

- Future work
  - Talk about features that would extend prototype
  - Revisit big vision



Your Team

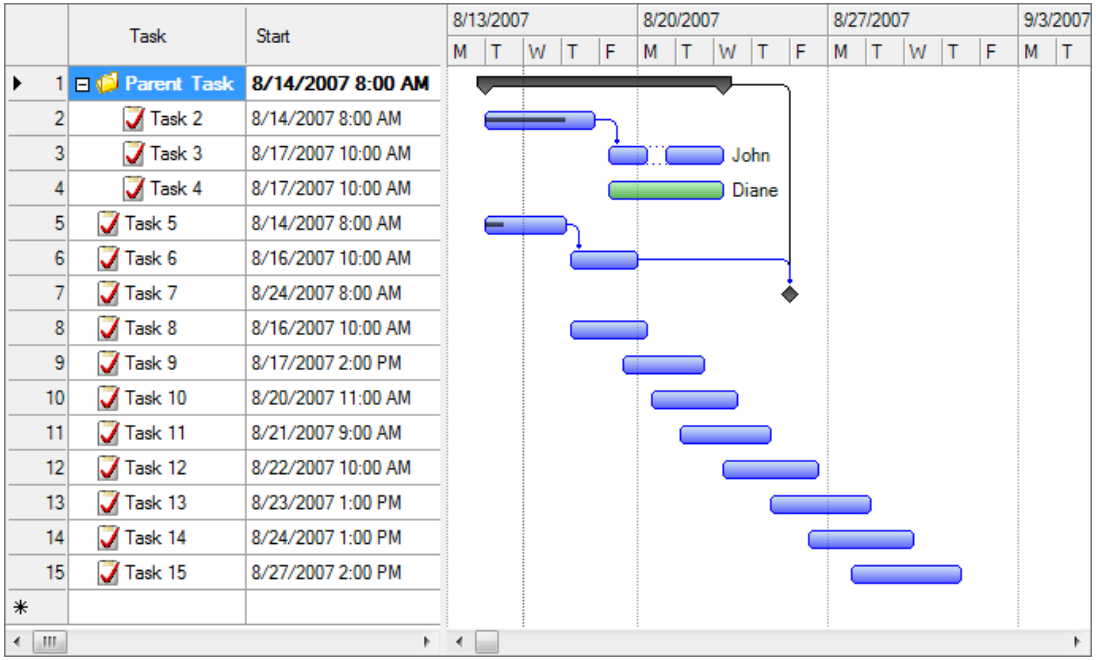
Some Team Tips

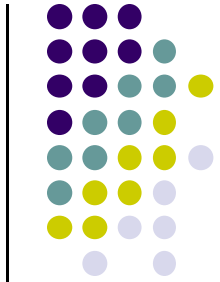


- You already have a team!
- Everyone (team members) doesn't have to do everything equally
- Team members can work on project aspects they are good at
- Example: Who is good at:
  - Android UI design (Android Studio design view, XML file, widgets, nice look)
  - Android programming (database, sensors, maps, backend)
  - Experimental evaluation/user studies
  - Machine learning
  - Writing, making presentations

Some Team Tips

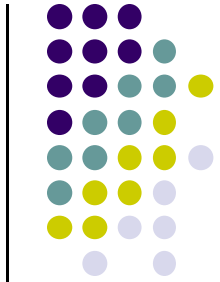
- Team should have an honest conversation
- Doing something different doesn't mean chilling
- Consider team online management tools, gantt charts, etc
- Assign tasks, mini-deadlines (every few days)
- Integrate features every few days => new version
- **Mantra:** Always have a working prototype, improve





Finding Idea to Work on

Pick an Idea to Work on

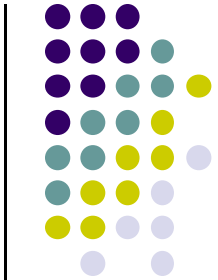


- Examples of previous projects from grad class:
  - Hearing aid
  - GeoChat
  - Finding car in parking lot
  - weather prediction along user's path

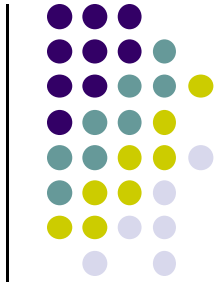
- Projects from Andrew Campbell class
  https://docs.google.com/document/d/1hg44pm9PPPnIxBfNthAktUD9XoHBLmkMdmq6BmJiWaI/pub

- What else is detected in ubicomp (5W's, 1H), examples ideas, how to do it in Android

Coming up with a Project

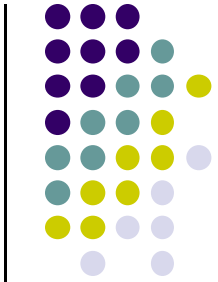


1. Search Android app market, what can you do differently
2. Click on papers,
    i. What areas you like?
    ii. What are your strengths? Machine learning? Signal processing?
3. Find papers you like within area or search Google Scholar, ACM digital library or IEEE Xplore
4. Can each paper be extended?
    a. Look at future work
    b. Repeat experiments + other things they didn't try. E.g.
        i. Re-implement a simple idea: E.g. Bewell
        ii. Implement PART(S) OF complex idea (e.g. place sense paper)
        iii. Propose new idea based on your prior knowledge/experience (GREAT!!! Maybe publishable?)



Resources

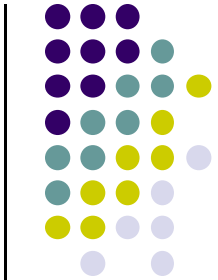
Resources



- I have set up a links page for mobile and ubicomp projects

http://web.cs.wpi.edu/~emmanuel/courses/ubicomp_projects_links.html

References



- Head First Android
- Android Nerd Ranch, 2nd edition
- Busy Coder's guide to Android version 6.3
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014