Game Time: Project Week

The Coding Bootcamp

Project Week Overview

You made it!!!

Project Week! (This Week)

Today's Class:
- Divide into groups
- Begin researching APIs
- Outline project ideas
- Submit Project Proposal for Approval
- Initial Design work

Next Class:
- Hardcore Development
- Instructor + TA Workshops

Saturday's Class:
- Hardcore Development

Project Week (Next Week)

Next Week (M/T):
•Hardcore Development

Next Week (W/Th):
•Hardcore Development
•Presentation Prep

Saturday's Class:
•Presentations!

Overall Schedule

- Day 1 (Today):
  - Students are introduced to Project Week, its goals, and requirements.
  - Students are introduced to git workflows
  - Students work with Instructors / TAs to identify feasible projects.
  - Students submit a Project Proposal
- Day 2:
  - Students learn how to use an agile workflow
  - Students are class time to work on Projects
  - Instructors + TAs work closely with groups to offer "code assistance"
- Day 3:
  - Students give micro presentations on their MVPs
  - Students plan their path forward
  - Students work on projects
- Day 4 (Next Week)
  - Students are given rest of class to work on Projects
- Day 5:
  - Students are given rest of class to work on Projects
  - Students begin planning their final presentations
- Day 6:
  - Students present their final project!

Teams

Groups

Team #1
Abdullahi Abdirashid, Olatunji Akanbi, Stephen Wood, Kenneth Bitzer

Team #2
Meng Vang, Brady Shinners, Hassan Lakis, Justin Trieschmann

Team #3
Abhinav Sharma, Tashi Wangmo Pierre Callies, Gamachis Yadesa

Team #4
Lindsay Lindner, Ray Becoskie Virginia de la Riva, Joe Hoffmann

Team #5
Karen Beltran Lopez, Nicole Brasaemle Corey Miller, David Steinmetz

Team #6
Rabic Ganni, Randall Olson Timothy Smith, Megerssa Tibesso Abdullahi Aideed

Team #7
Duncan Moore, Robert Phillips Sam Elsola , Andrew Vala

Task

Coding Requirements

- Must use at least two APIs
- Must use AJAX to pull data
- Must utilize at least one new library or technology that we haven't discussed
- Must have a polished frontend / UI
- Must meet good quality coding standards (indentation, scoping, naming)
- Must NOT use alerts, confirms, or prompts (look into *modals!)*
- Must have some sort of repeating element (table, columns, etc)
- Must use Bootstrap or Alternative CSS Framework
- Must be Deployed (Github Pages)
- Must have User Input Validation

Coding – Nice To Haves

- **Utilize Firebase for <u>Persistent Data Storage</u>** (Consider this **<u>basically a requirement</u>**).

- **Mobile Responsive**

- **Use an alternative CSS framework like Materialize**

Presentation Requirement

- You will also be responsible for preparing a 10 minute presentation.
- This will be a formal presentation.
- One in which you explain in detail:
  - Your overall application's concept
  - The motivation for its development
  - Your design process
  - The technologies you used (and briefly how they work)
  - A demonstration of its functionality
  - Directions for future development
- Treat the presentation seriously!
- Talking intelligently about tech > doing tech sometimes.

Metrics

Metrics

- Concept
- Design
- Functionality
- Collaboration
- Presentation

Awards Yay!

- Most Awe-Inspiring
- Most Useful
- Most Creative
- Best Use of Tech
- Best UI/UX
- Most Hilarious
- Most Disruptive
- Most Socially Conscious

API Suggestions

API Suggestions

Stick to APIs that do all of the following:

- Simple or no authentication
- JSON response returned
- Well documented

API Suggestions

API List

Name	Allows CORS	Authentication Method	Authentication Required
Spotify Web API	yes	oauth	yes/no
reddit API	yes	oauth	yes
Youtube API	yes	API key	yes/no
Linkedin	yes	API key	yes
Soundcloud	yes	API key	yes
Wikipedia	yes	n/a	n/a
Rotten Tomatoes	yes	API key	yes
flickr	yes	oauth	yes
Pinterest API	yes	oauth	yes
Google Custom Search API	yes	API key	yes
Tumblr API	yes	oauth	yes
Marvel API	yes	API key	yes
GeoNames API	yes	username	yes
OMDB	yes	n/a	n/a
Zillow	no	ID	yes
Open Street Maps	yes	n/a	no
GovTrack.us	yes	n/a	no
Sqoot	yes	API key	yes
Google Maps	yes	API key	yes
Giphy	yes	API key	yes
JSFiddle	yes	n/a	n/a
Bitcoin Charts	yes	n/a	n/a
StackExchange API	yes	oauth	yes
Pro Publica Non Profit API	Yes?	n/a	no
U.S. City & County Web Data API	Yes	n/a	no
UN Data API	Yes	API key	yes
Face++ API	Yes	API Key	Yes
MusiXMatch	Yes	API Key	Yes
Words API	Yes	Token	No

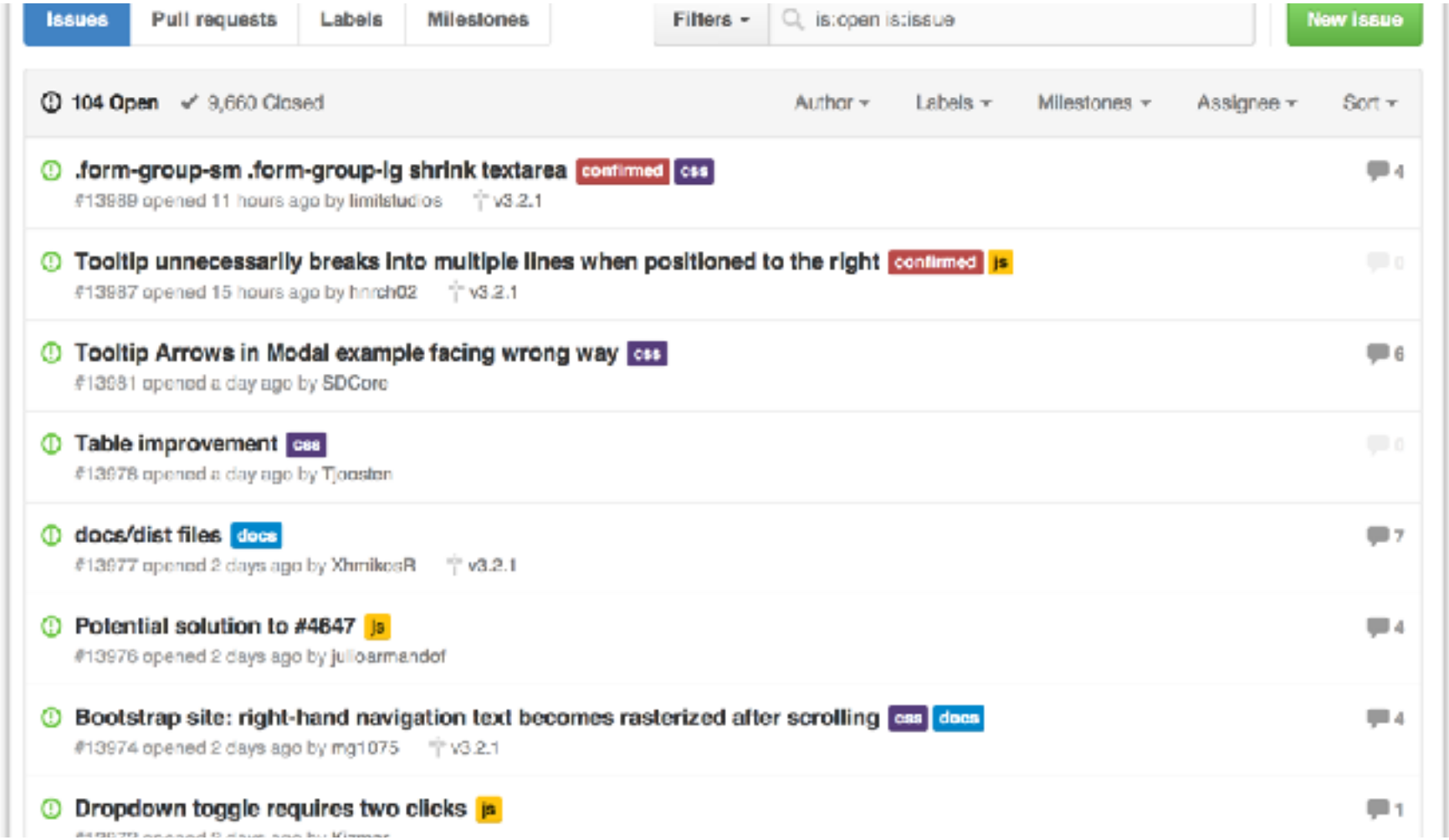
Tools

Collaboration is Critical!

- Steering a project with remote developers like this one can be challenging.

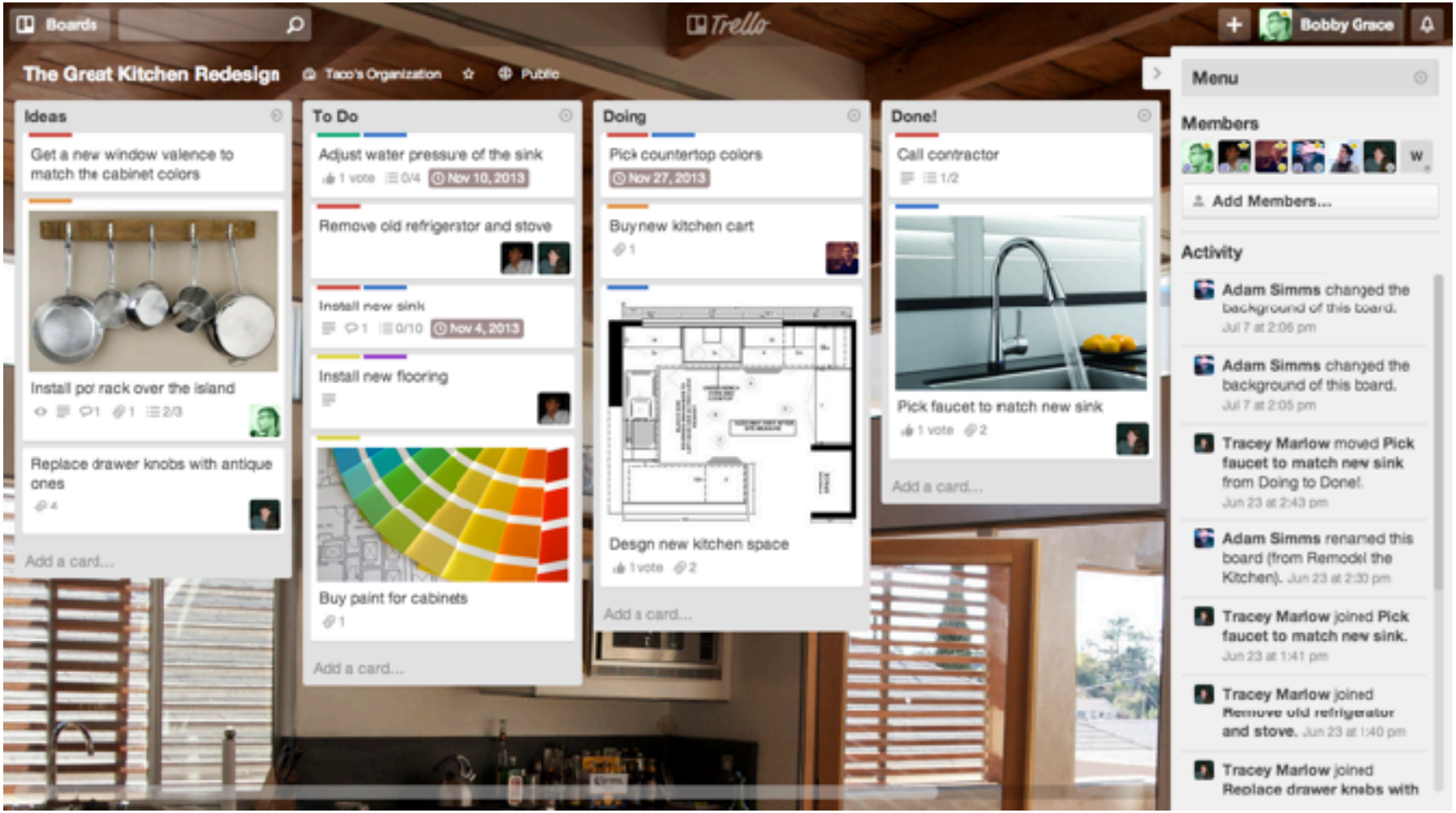
- Consider using the following tools…

GitHub Issues



- GitHub Issues are a great way to keep track of bugs, feature requests, etc.

Trello



- Trello is a great project management tool for creating to-do lists and communicating updates.

Example Project Ideas

Event Searcher

•Users type in the name of their favorite sporting team.

•Then using the SeatGeek API, your web application points them to the next game and location where the team is playing.

•The web application also provides them a link to Ebay where they can purchase memorabilia associated with the team.

UN Data API

•Use the Unofficial UN Data API to search a user specified country's health records.

•Then use a secondary data source (Google Maps, Flickr, YouTube, etc.) to provide additional context or data.

•Examples: Per Capita Govt Expenditures on Health, # of Physicians, Deaths due to HIV, Malaria Cases etc, low birth weight newborns

Facial Recognition

- Use the Face++ API

- Allow users to provide a URL link to a facial image then provide viewers with information on the image's gender, race, whether they are wearing glasses, and their age

- Use Firebase to update ALL users of the site about the current search

Or…

Just do your own thing.
Be creative! Be ambitious!

Today's Focus

By End of Day - Today

- **Github Respository Setup**
- **Brainstorm possible ideas**
- **Begin API Research**
- **Create an initial draft / sketch of the final design**
- **<u>Create a short 1 page proposal listing out each of the following:</u>**
  - **Project Title**
  - **Team Members**
  - **Project Description**
  - **Sketch of Final Product**
  - **APIs to be Used**
  - **Rough Breakdown of Tasks**

Create a Repository

* Once group member should create a new Github repository. Don't worry about the project name now, this can be changed later.

* From the repo's main page, click the "Settings" tab.

* Once in the repo's settings, select the "Collaborators" menu item on the left.

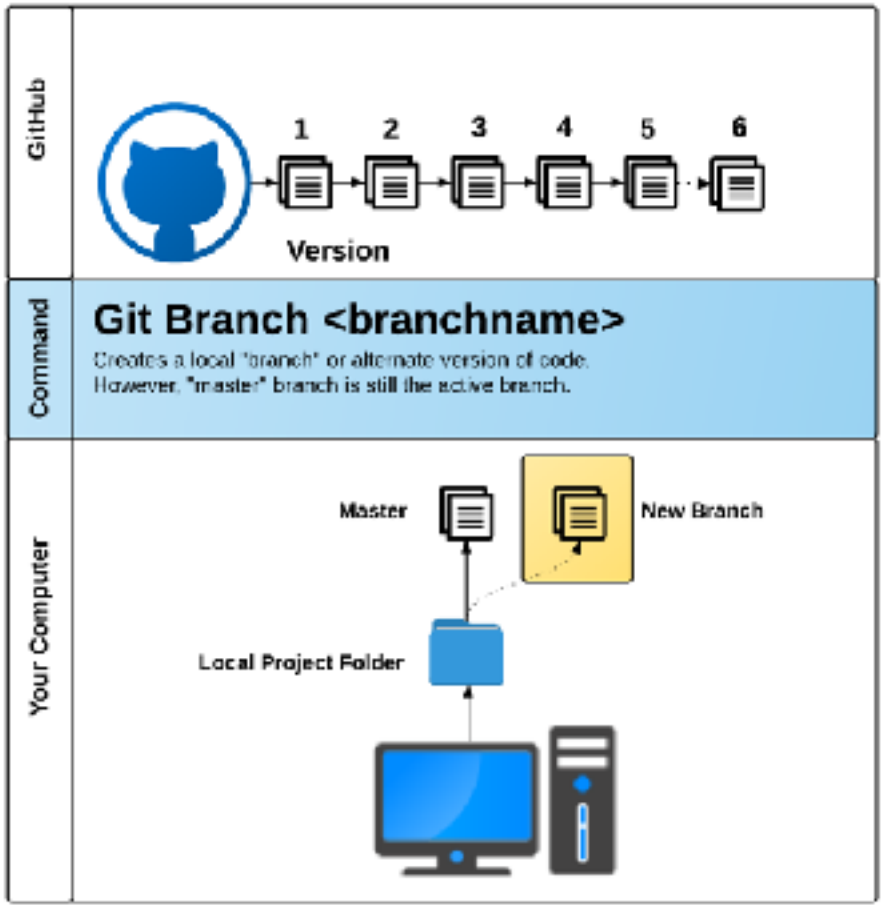
* From the "Collaborators" page invite your group members to be project collaborators by entering their Github usernames one at a time.

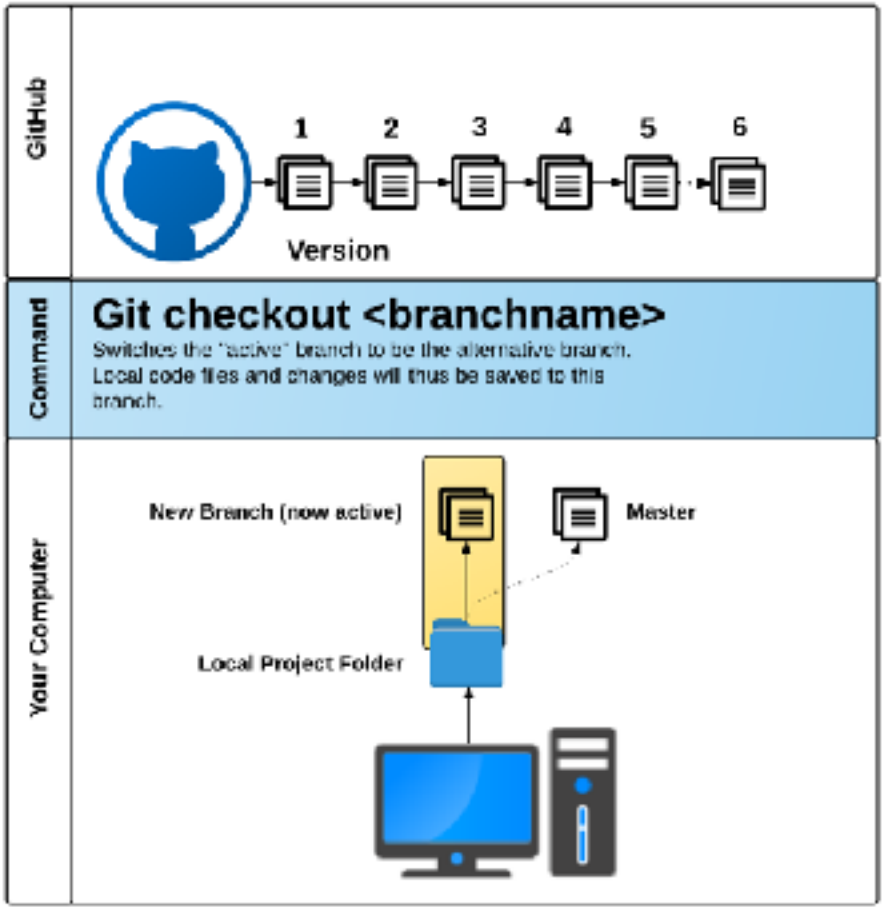
* Each invited group member should receive an email they must open to accept the invitation.

* **Hints**:

* Ask an instructor or TA if you get stuck!

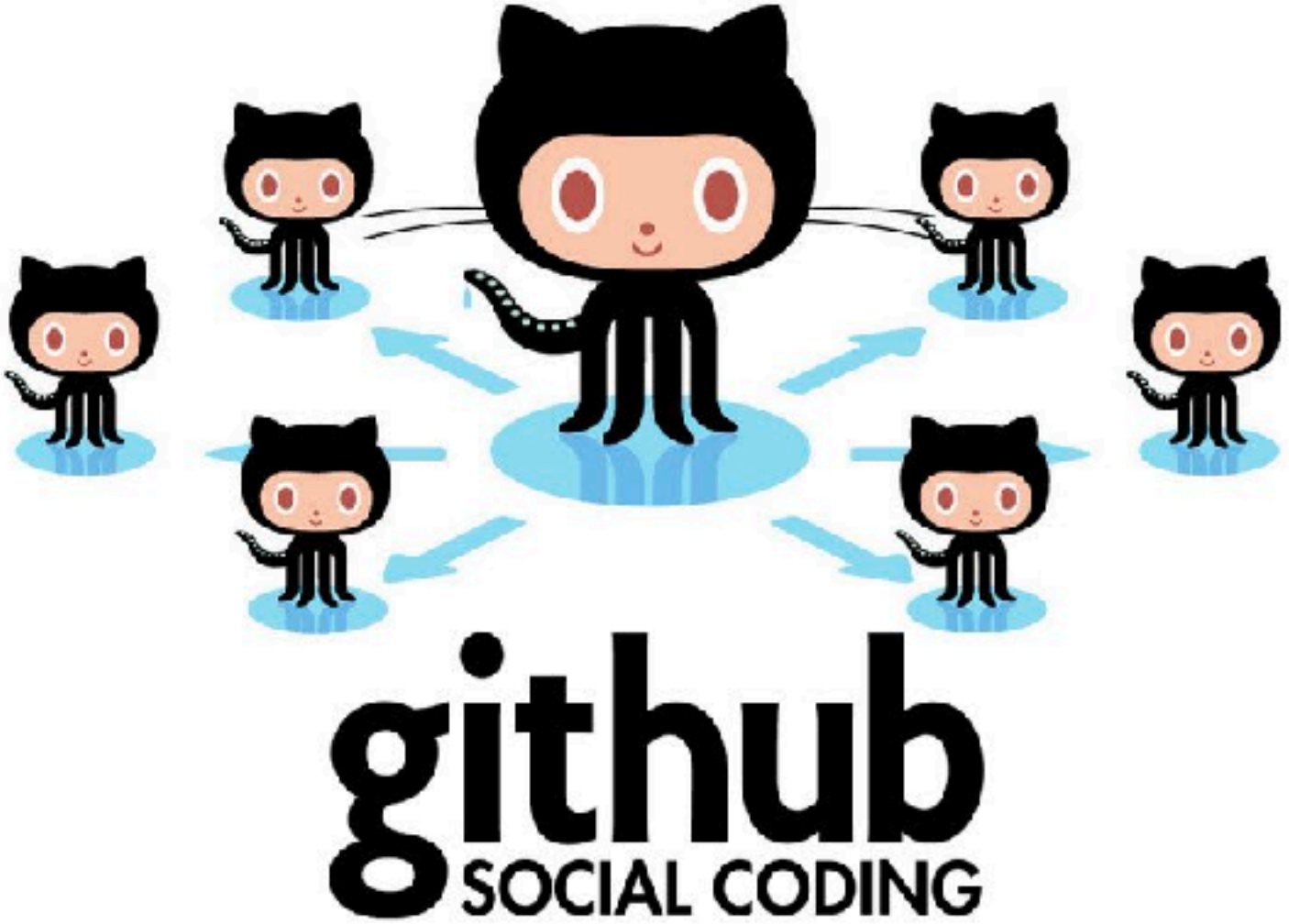
GitHub Pull Requests





•GitHub Pull Requests are a great way to “combine” code when multiple users are working on the same files.

Pull Requests and Code Review



Protect Master Branch (5 mins)

* Only one member per project group needs to complete this activity.

* Navigate back to the repo's "Settings" page and then select "Branches" from the left sidebar.

* Under "Branch Protection Rules" select "master" from the dropdown.

* You should be presented with some options, check off the following:

  * "Protect this branch"

  * "Require pull request reviews before merging"

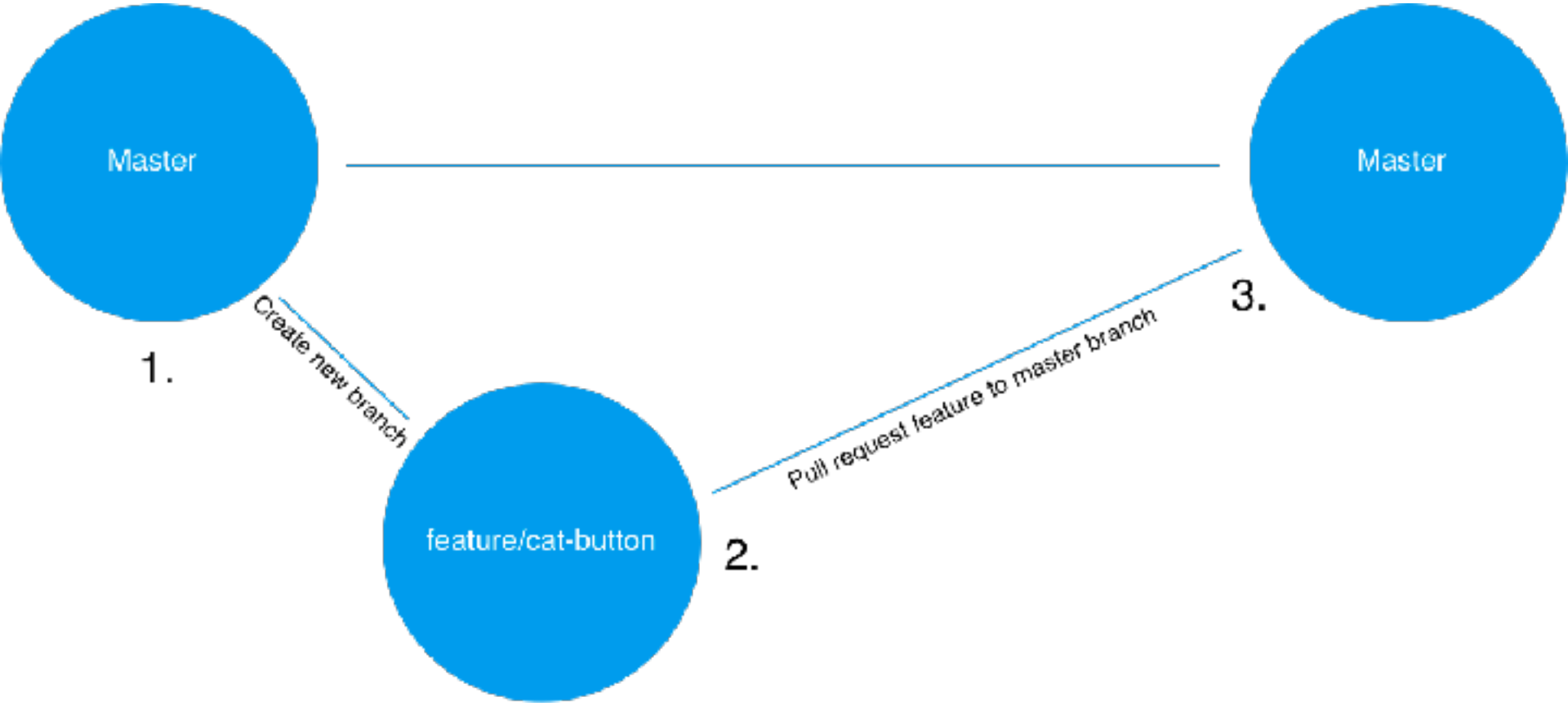
  * "Include administrators"

* If completed successfully, no one should be able to push directly to the master branch. Instead, all changes must be made in the form of pull requests that are to be reviewed by another group member.

* **Hints**:

* Ask an instructor or TA for assistance if you get stuck!

Branching


Master
Master
1.
Create new branch
feature/cat-button
2.
Pull request feature to master branch
3.

Everyone Do: Git Branching/Pushing

- **Part I**: Branching and Submitting a Pull Request

- In this section we will create a branch, add a feature, and submit a pull request. **Only one group member should complete this section, everyone else should observe.**

- **Part II**: Reviewing a Pull Request

- In this section we will review the pull request from Part I and merge it into master. **A different project member should complete this section while others observe**.

Students Brainstorm

- Work with groups to identify ideas, research APIs, and create project designs.

- Instructor/TA should begin hosting "workshops" to help steer groups in the right direction.

BREAK TIME

BACK IN 15

Students Brainstorm

- Work with groups to identify ideas, research APIs, and create project designs.

- Instructor/TA should begin hosting "workshops" to help steer groups in the right direction.

Questions