Gary
APCS2 PD8

KDubs EDubs
KimJoshUn

Project Proposal

Name of Project: Twitter Tree
Use Processing to visualize and sort Twitter topics by popularity and time posted.

The user will be presented with a visual interface that allows them to select from a number of 'bubbles,' each of which contains a different 'trending' topic. When clicked on, the bubble will expand into a *binary tree* that sorts tweets on the aforementioned topic based on popularity and time. The left branch will contain the most recent tweets, updated in real time, while the right branch will contain the tweets deemed most popular, based on likes and retweets.

Twitter data will be extracted by using pre existing Java libraries for the Twitter API:
- http://twitter4j.org/en/

Structure:
- Node based tree (or should we use an arraylist)
    - Each node will contain an object that represents a tweet

Classes/Methods:
- Cloud - initial display of the trending topics
    - clickedOn() - when user clicks on bubble, call a new Tree to display tweets of said bubble
    - search() - maybe allow user to search for any topic on twitter, regardless of if it is the trending or not
- Tree - sorted tree that contains specific tweets based on recency and popularity
    - rearrange() - allows user to rearrange the tweets within the tree (maybe idk about this tho)
    - remove() - allows user to remove a certain tweet and replace it with the next relevant tweet
        - Similar to how elements shift up in binary trees

Prioritized To-do list:
- Complete project proposal
    - What type of tree will we use?
    - What is our timeline?
    - Complete to-do list
- Create main classes
- Create encapsulation for tree to contain data
- Implement twitter4j library to work in conjunction with interface
- Complete basic GUI

        - Have the circles have them expand when clicked on, etc.
- Test each method*

Timeline

V1:
- Implement main classes and methods
- have basic GUI done (without data)

V2:
- Integrate twitter data into data structure

V3:
- Display data structure in GUI

V4:
- Add additional methods

Other Ideas:

Escape the School

Using processing to create a game where the goal is to escape Stuyvesant High School.

Upon generation, the character will be placed on the 10th floor with the task or finding their way out of the school.

Can't move backwards, the floor will break after you leave a “square.”

On each level, there will be different clues. All clues must be found to complete the puzzle to proceed to the next floor.

Classes & Methods:

- Student
    - clickedOn(): will check to see if there’s anything hidden in the spot and move there.
    - displayInv(): Show what you have in your bag
- Floor
    - hide(): Hide all the items in the floor
    - check(): Upon exiting, check if the floor is complete
    - break(): break the floor after character exits

List of Craigs

Display in Processing a forum similar to Craigslist. Our project differs, in that instead of buyers looking for sellers, buyers will make a posting with their request, and sellers are able to match with a buyer.

Key features: ability to write postings as buyers and sellers in Java. Ability to view most popular products/categories.

Utilize a max-heap to show sellers top prices buyers are willing to pay.

Classes and Methods:

- Buyer
  - createListing(Category): Creates a listing for a product, with specified category
  - setPrice(int): Sets buy price
  - approveMatch(): User can approve or deny a purchase, approved so long given price from seller is less than or equal to buy price
    - Maybe allow user to be removed from heap so as to not be badgered by seller?
- Seller
  - requestPrice(Category): Request
  - findInDemand(): Finds the most in-demand category (utilize a 2D array)
  - matchRequest(ID, price): Requests a transaction with the specified user for the price given

---

Line Cutters
Using processing to create a game where the goal is to get to the front of the line.

Obstacles will block the way. You can only move forward when the spotter is not looking. If you are spotted by the spotter you are taken out of the game.

Classes & Methods:

- Player
  - Move(): allow you to move in the direction you're facing
  - isCaught(): If the spotter inspects you it will return true
  - converge(): Go back into the line
- Spotter
  - Inspect(): Look for a random number of seconds
  - Move(): Move to a different section of the line
  - Turn(): Change direction

TD Balloon
Using processing to create a game where the goal is to protect your headquarters from the invasions of balloons.

This is like the game from NinjaKiwi. The balloons follow a path and you have to place monkeys that pop the balloons. The balloons are queued in one way and then all travel down along a path.

Classes & Methods:

- Popper
- Enhancer

Bop It
Using processing to create a game like bop it.
There would be a random generation of patterns and the person needs to follow the same pattern that's given.
