Meeting Notes: January 28th, 4p

Present library functionalities

Everett: kivy

experimentation with image resizing, sound elements, 3d monkey faces…

plenty of tutorial resources

joining images, play videos

limitation: changing shape of widgets to circles would take a little maneuvering

next meeting: prototype w/Wiki picture in the frame

look for a UI with a more custom feel to it?

focus on a simple UI at first, work on backend

look at kivy library dependencies

TODO: Sarah, Ashley download kivy

need to run text parser and UI creator in parallel

Sarah: BeautifulSoup

can create trees from each HTML tag

useful for pulling and separating hyperlinks and image links

Ashley: networking very easy and straightforward w/libraries

mini-challenges in optimization

how many connections? in parallel?

would multi-threading improve performance?

igraph prototype: basic node structure with several different levels

can save all graph attributes to a data structure

many different graph algorithms

visualization element completely detached from graph data

C library w/Python interface

graph could look more organic, with branches with randomized lengths/angles

from parent node

could be an issue w/child nodes

igraph could possibly be too heavy-duty in conjunction with kivy widgets

do we need graph data to send to UI?

Homework 2:

Everett: Use cases for kivy

each widget broken into classes

Break Task #1 into each specialty group

Ashley & Sarah: use cases

Sarah: UML

Use Case diagrams-> give to one person

State diagrams

lots of loops when adding nodes

big picture states, get there through different actions

divvy up each type of diagram

class diagrams: everybody

use case diagrams: Everett

sequence diagrams: Ashley

state charts: Sarah

Task 1: progress done by 2/6 (at least halfway done)

next steps: Everett: stress-test kivy w/lots of images to determine feasibility =>prototype, proof

of concept

-one version with local images, one version with online images

Ashley: network optimization, install kivy

everyone: outline of required classes, unit testing, get everyone’s libraries, post sample code

discuss github branches (see research notes link)

testing: Python library for unit tests

assert statements

see modules working together

individual white boarding

can divvy up sub-tasks as they present themselves