Exploring Computer Science Curriculum notes for designing the "Apps Class"

Exploring CS Curriculum: http://www.exploringcs.org/curriculum (Version 5.0 under review)
Apps for Good Curriculum: http://dl.dropboxusercontent.com/u/155442535/Educator%20Zone/index.html
The Target Goal: Project Guidelines and Criteria
http://dl.dropboxusercontent.com/u/155442535/Educator%20Zone/Competition%202013%20-%20Submission%20Guidelines%20and%20Judging%20Criteria.html
Sample of Apps of Finalist Project details: http://www.appsforgood.org/public/finalist-teams

Purpose Statement:
Even though I have been teaching Exploring Computer Science (CS) for two years without using the formal exploringcs.org curriculum, mine follows the first four units fairly well. My Exploring CS class is only one semester, so I only achieve the first 18 weeks / 4 units of the class, ending with programming in Scratch. For the Apps Class, we will be integrating the Exploring CS material with the Apps for Good assignments, so the class will become a specialized version replacing sections of the Exploring CS units with the more team design focused lessons of Apps for Good and likely culminating in using App Inventor as the final programming language, although it still may be considered to start students in Scratch. I am gathering my notes and ideas, plus some key points to help in understanding the Exploring CS material and how to edit it for our new Class.

For reference, the major Units for the full Exploring CS class are:

1. Human Computer Interaction (HCI - 4 weeks)
2. Problem Solving (PS - 4 weeks)
3. Web Design (WEB - 5 weeks)
4. Introduction to Programming (PR - 6 weeks)
5. Computing and Data Analysis (DA - 6 weeks)
6. Robotics (ROB - 7 weeks)
7. The Societal Impacts of Computing (woven throughout the course)

Proposed Guidelines:

1. **Exploring CS Unit Focus 1-4:** This is only a semester class, only material in units 1 through 4 and integration of "7.Societal impacts of computing" to be "woven throughout" are to be included, this is the first 20 weeks of instruction or so. At Garfield, our next Introduction to Programming class, covers much of Computing & Data Analysis unit and delves into further concepts of computer science (we used to use Scribbler robots, but have not this last year.)
2. **Reduce Exploring CS material:** To integrate the focus on designing Apps for good, we will need to reduce the material from the Exploring CS curriculum, making sure to keep areas most relevant for creating effective & helpful Apps.
3. **Paint.Net – how to learn a new application:** I normally have a unit on Paint.NET (an open source PC photo editing & graphics software) to teach the different ways to learn a new application so students develop the approach that works the best for them. Paint.NET will likely work for the Apps class since we'll need to create graphics; then students will have several more opportunities to learn the variety of new tools in the class including App Inventor and Balsamiq.
4. **Website Design:** Website design needs to be included since it is key to promotion online and very helpful in creating & updating information pages for an App. However after teaching HTML & CSS, we may want to use Google sites since they are included with a Google Account, which is inherent to App Inventor. *Note: we will need to determine how best to*

protect students' security when posting material online; perhaps a general Google project account can be created for each so it is not linked back to individual student identities.

5. **Use Website Design to introduce User Experience:** Website development is a good place to introduce Site Maps and Wireframes by example and perhaps even have students start to use Balsamiq. Simple website design may be more tangible to teach this process and introduce usability and good design. Then when students start on their app design, they will have a better idea of where they are heading. In fact it may work to introduce creating Personas & their goals as well here. I am considering an entire class website project – teams of students contributing the pages for a school-wide project.
6. **Scratch vs App Inventor for first Programming Language:** I have had great success using Scratch to introduce Programming Concepts (see link below) and I wonder if for students that have not programmed if we should start with Scratch before going on to App Inventor – time is the concern of course. If each team has at least one or two students that understand these concepts (i.e. have taken a programming class) we may be able to go directly to App Inventor as long as those students can help others through these concepts. Scratch Programming Conceps: http://www.garfieldcs.com/wordpress/wordpress/wp-content/uploads/2011/12/Scratch-Concepts.pdf
7.