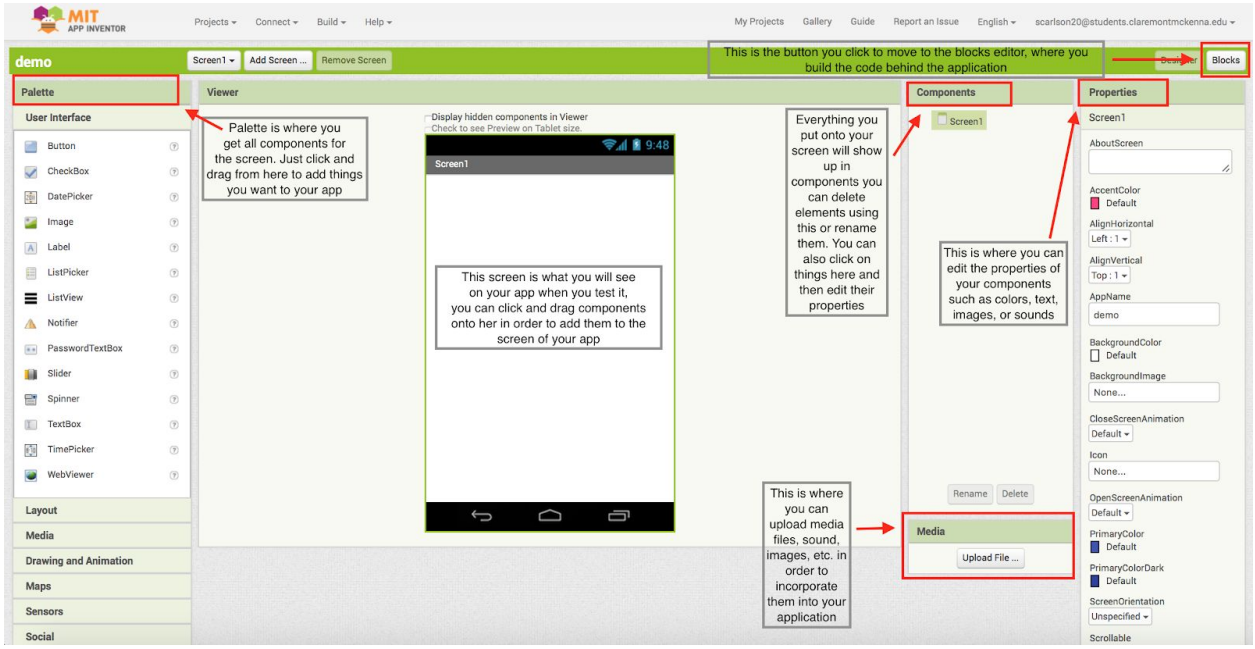
How to Teach AppInventor Curriculum

Part 1: All About the Platform

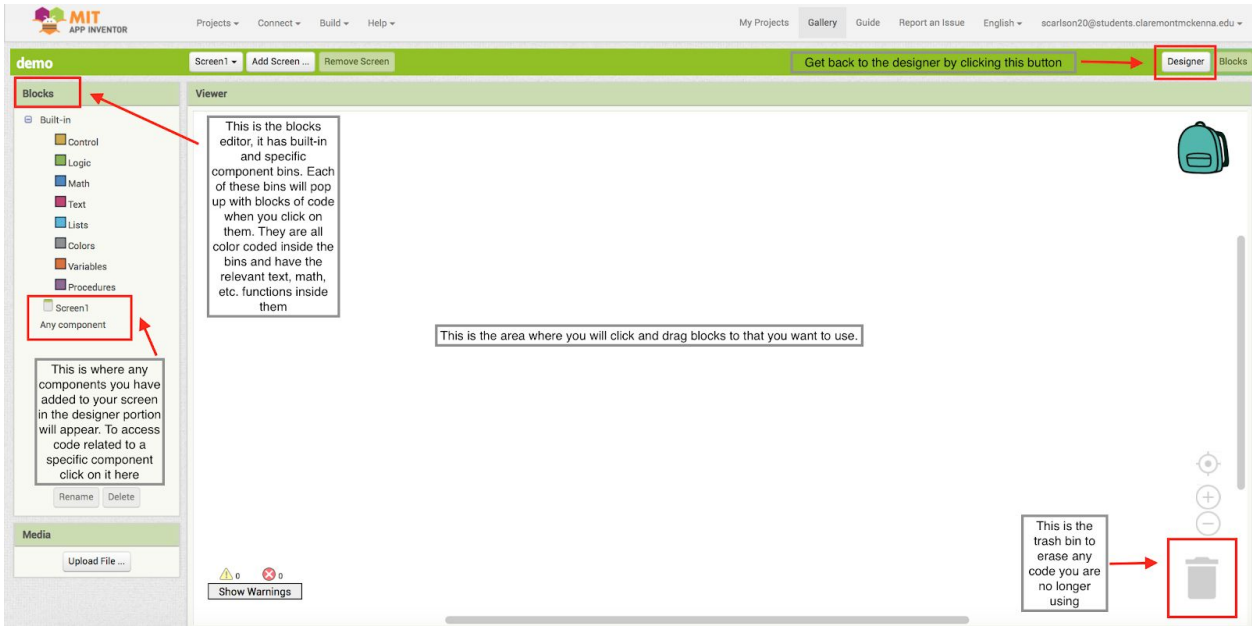
MIT's AppInventor: http://ai2.appinventor.mit.edu/

Basics:

- Have student log in using a gmail - most should have a school account but if they don't they can create a gmail (pretend to be at least 13, if they don't the first time use an incognito window to create one)
    - If you create a gmail make sure you document students username and password somewhere as they often forget
- Designer Screen - where you design the interface of you application



- Blocks Screen - where you code functionality behind your application



- Testing on AppInventor
    - You can test your AppInventor applications on android phones
    - Instructions on the application and how-to are here: http://appinventor.mit.edu/explore/ai2/setup-device-wifi.html
    - Quick troubleshooting - make sure they are on the same wifi network, if it still isn't working try to reset the connection (under connect dropdown menu) and try again
    - If nothing works, you can download the emulator onto your computer and connect with a USB cord, but this is last resort as it take a while to download and requires restarting the computer.

Part 2: Tips for Teaching Coding

Goals: Build their confidence and self-sufficiency

- **Hands off teaching**: Don't take over a task for a student if they're frustrated. Just be patient and do your best to explain. Even if it is something little let them use the keyboard and mouse for everything.
- **Focus on the learning, not the project**: The most important goals of this class are to teach problem solving, growth mindset, creative confidence, and to instill a comfort with tech. As such, the learning process of building the projects is far more than the projects themselves. Emphasize the process - overcoming unexpected problems, communication, exploration etc - over the quality of the final project.
- **Explore**: If a student wants to try something, even if you know it won't work, let them (as long as it's safe, of course) and be a partner in exploration. If you don't know the answer to something, don't be embarrassed - make it a learning opportunity. Work with the student to figure out the answer together.
- **Have them help each other:** If you have a student who understands a concept well, encourage them to teach it to another student who might be struggling. This not only reinforced the ideas for students teaching them, but also makes things easier as the teacher.
- **Let them learn by doing:** Work to find the balance of letting students figure out things on their own and giving them guidance. I've intentionally left my students alone before because I knew they could figure it out on their own, and they were growing too reliant on me
- **Check for understand, not just completion:** Make sure you check to make sure a students understand before moving on. Don't assume a student understands something just because they say they do - evaluate them by asking questions or for them to explain a concept to see where their understanding is. Concepts will build on each other so understanding at each step is important

Part 3: Ideas for how to teach AppInventor

Here is a site that has tutorials and overviews for all AppInventor projects listed below. It also has code.org activities for students who have finish early and have extra time. If I am giving

tutorials, I usually give the kids the link to this and have them go to tutorials and code.org from there as many of them can't type in URL's without a few mistakes.
https://scarlson20.wixsite.com/guapps

Overview v. tutorial

- Overview - created by StreetCode
    - These are step by step instructions through all StreetCode AppInventor tutorials
    - Benefits - very easy to read + help mentors step kids through a project
    - Downside - often not good to give directly to kids as they have screenshots of all the blocks and the screen and no pictures to help guide them through, some students have settings on their school google accounts and cannot access these, as they are a part of another organization's google suite.
- Tutorial - Created by AppInventor
    - These are step by step instructions through all AppInventor tutorials
    - Benefits - Have clear pictures and very in depth descriptions of every step of making the application, often have starter apps with preloaded media files which can be very helpful
    - Downside - vary a lot in quality and format, also many of the apps were adapted or expanded upone by streetcode and they don't have a tutorial that directly links to them
        - You can teach other apps that AppInventor has tutorial for, you don't have to stick to StreetCode tutorials

Two Strategies for teaching:

- Give them the tutorials
    - Benefits - you don't need much mentor attention, often younger kids who can't figure out how to do things on their own benefit from this
    - Downsides - less active learning as they can copy down a little from the tutorial but usually they still get a lot of growth, after PaintPot MIT's tutorials are much less helpful.
- Have mentors use overviews to give them each step of the tutorial as they need it
    - Benefits - much more learning by doing
    - Downsides - need much more mentor attention, slow to start before they have a good grasp on AppInventor

Part 4: Tutorial Order and information

Below is the list of tutorials we used (there are many more on AppInventor if you are interested in checking them out.) in the order in which we teach them. The links below them are to the StreetCode "Project Bundles" which contain the overview, and .aia files for both the basic and expanded versions of the App. If you upload these files to App

Inventor it will be the full designer and blocks for the application as described in the Overview.

All tutorials and overviews: https://scarlson20.wixsite.com/guapps/all-tutorials-overviews

1. MLK
This application is a button which when clicked plays MLK's I Have A Dream Speech. If students finish MLK they can then add a Malcolm X Button. This is a really easy app that has a fun result of the kids (talking to them) it is a good introduction because the blocks are simple but the interface you build looks exciting
https://drive.google.com/open?id=1DurbBwEBgHhQYwexljjx1zsYGdxMy9MV

2. Magic 8 Ball
This application has an image of a Magic 8 Ball and when students shake the phone, it will say common Magic 8 Ball responses. This app teaches them how to make a phone play specific text of their choosing, and allows them more creative freedom as they can come up with the responses the phone will say.
https://drive.google.com/open?id=1SFICK4Pm6UEisO73N2I6iZCbg_RmJM69

3. Paint Pot
In this application students use a canvas to create an application where they can draw in different colors. The expansion also teaches Students to incorporate the camera into the application.
https://drive.google.com/open?id=10aB2Q_Pv8bs7wNReL0rz3KrFCLRvJ_dj

4. Ball Bounce
This application has a ball that moves around on the screen based on where you "fling" it. It teaches kids how to work with animated objects.
https://drive.google.com/open?id=18u3F85RmPaltr0gR6aeEU3UGTAz6g6Ww

5. Space Invaders
This application has a rocket ship that shoots balls at a flying saucer. It teaches students to keep score, use timer and clock components, and deal with collisions of animated objects.
https://drive.google.com/open?id=1xdUcr-kEAHuITSCEoJu4flWQNQWwEMhB

6. Mole Mash
This application has a mole that pops up randomly around the screen and students expand upon their use of clocks, collisions, and animation from Space Invaders.
https://drive.google.com/open?id=1HSI4gmK_vBwsaQ36FxTouhreWp-l-_QO

7. Pirate Ship

This application has a ship that collides with coins. It continues with the use of collisons, clocks, and animation.
https://drive.google.com/open?id=1FBPaSbynfM97j9rjGB-TB3f-b0SCj0K2