Visual Programming with Scratch

Session Plan

- Presentation: Overview of Visual Programming & Scratch (~10 minutes)
- Activity: A hands-on introduction to Scratch (~30 minutes)
- Presentation & Demonstration: Some tips for introducing coding into your classroom (~20 minutes)

Visual Programming Languages

- Languages that allow you to Code by using visual elements, e.g. dragging and dropping blocks
- Examples include:
  - ScratchJr
  - Scratch
  - LEGO Mindstorms
  - AppInventor

Scratch

- Who has heard of Scratch?
- The Philosophy behind Scratch
  - Designed for *tinkerability*
  - Encourages collaboration and sharing
- Low floor, wide walls and high ceiling
- Who is already using it in their classrooms?

Scratch in K - 12

- Commonly used to introduce K - 12 students to Coding and Computational Thinking
- Students can create games, animations and interactive stories
- Quality resources and lessons available online (see CS 4 CSO website)

Scratch Activity (~30 mins)

- Go to the Scratch website: www.scratch.mit.edu
- Work through the Introduction to Scratch handout
- If you have already finished the Introduction to Scratch handout, I have another tutorial you can work through

Coding in your Classroom with Scratch: Some Tips

Background

- A Stage 2 coding class at local public school (~25 students)
- Mix of Gifted and Talented (GATS) Year 3 and 4 students
- Nominated for class by teacher or parent
- Computer lab with ~30 netbooks
- Introduction to coding with Scratch
- No formal assessment as part of the class (1 hour lesson)

Schedule

- Week 5: Account sign-up & intro tutorial
- Week 6: Continued intro tutorial
- Week 7: CC Snowball Fight tutorial
- Week 8: CC Archery tutorial
- Week 9: CC Flappy Parrot tutorial
- Week 10: Sharing session

Some Positives & Some Hurdles

Positive #1: Tutorials

- Tutorials developed and shared by Code Club
- Step-by-step (and usually straightforward) instructions
- Some (e.g. Flappy Parrot) allowed for some customisation
- After a few of these I think students could move onto creating their own ideas

Positive #2: Collaboration

- With 25 students the teacher and I were very busy
- Some students acted as helpers
- Groups asked each other questions when they got stuck
- Teacher is hoping that some will act as experts when introduced to other students

Positive #3: Sharing

- Last week of the class was a sharing session
- Students (around a third of the class) shared:
  - Their own creation
  - A modified version of a CC tutorial (e.g. Flappy Hippo)
- Students fixed *bugs* (mistakes in code) as a group

Hurdle #1: Accounts & Passwords

- Each student set up their own Scratch Account in Lesson 1
- Teacher instructed students to use certain pattern for username and password
- Took a fair bit of time for everyone to register
- A few students would forget their username/password most weeks
- **Register for a Scratch Teacher Account!**

Hurdle #2: External Resources

- Some Code Club tutorials require external resources
- e.g. students need to access *jumpto.cc/archery-resources*
- Could have demonstrated explicitly how to access these
- Small issues but takes time to solve
- May be better to stick with built-in Scratch resources for beginners

Hurdle #3: Remixing & Plagiarism

- All shared Scratch projects can be remixed
- All blocks, sprites, sounds etc can be copied
- Students presented remixed project as their own in sharing session
- Remix tree shows original project but not what has changed
- Could be a challenge when assessing (no assessment in this class)

Teacher Account Demonstration

Please complete the Scratcher Sign-Up form

Link @ cs4s.github.io/cso-2016 on the Visual Programming page