SCRATCH
CREATIVE COMPUTING

Prepared by Bronwyn Welch
CMA Conference 2014
Today’s journey

- Background
- Big ideas of programming
- Creative computing
- The SCRATCH environment
- Complete a small task
- Tasks, projects, sharing and showcasing
- Case studies
- ScratchEd & support
- Reflection
Background

- Context (2009)
- Makers vs Users (2012)
- https://www.youtube.com/watch?v=nKlugyen5nc
- Working as a Mathematician – Problem Solving
1. Find an interesting, meaningful or worthwhile problem to solve...

2. Informally explore, experiment, collect data

3. From patterns in the data create conjectures, hypotheses, theories

4. Use toolbox of problem solving strategies to prove or disprove theories

5. Use toolbox of basic skills

6. Extend or generalise, what else can we learn?

7. Publish / Communicate

8. Go to Step One

Being a Mathematician
The Big Ideas of programming

- Think: Recipe
- Connecting computing – impacts/connections
- Developing artefacts – problem/idea
- Abstracting – modelling problem/solution
- Analysing problems and artefacts – evaluation/debugging/efficiency
- Communicating – Explain the process/meaning/summarise
- Working effectively in teams – components tasks & collaboration
Creative Computing

Creativity: Computing is a creative activity.

A. Computing fosters the creation of artefacts.
   1. Computing enables people to create digitally—including creating knowledge, tools, expressions of ideas, and solutions to problems.
   2. Computing enables people to translate intention into digital artefacts.

B. Computing fosters creative expression.
   2. Computing fosters the creation of new forms of expression.
   3. Computing enables creative exploration that informs and inspires.

C. Programming is a creative process.
   1. Some programs are developed to satisfy personal curiosity or for creative expression.
   2. Some programs are developed to solve problems, develop new knowledge, or help people, organizations, or society.
SCRATCH - the environment

- [http://scratch.mit.edu/](http://scratch.mit.edu/)
- Google “Scratch” or go to [http://scratch.mit.edu/](http://scratch.mit.edu/)
- Scripts, blocks, loops, controls, sprites
Let’s code!

- Motion
- Sound
- Control
- Pen
- PLAY

Your task: Low floor, high ceiling

- Create an equilateral triangle using the four types of scripts demonstrated
- For fast finishers:
  - Change the colour or create your own sprite
  - Create another triangle, inverted, exactly over the top of your first triangle
  - Can you have your script start and finish at (0, 0)?

- What were your challenges? What did you learn?
Ways to use SCRATCH in your classroom

- Learning
- Consolidating
- Collaborating
- Making
  - Presentations
  - Whatever you want!

- Sharing learning
Case Studies

- A – engaged & then wanted to be a mentor
- E – wanted to create and solve problems
- Z – wanted to make the most frustrating game on earth
ScratchEd- help & share

- http://scratch.mit.edu/educators/

- Maths tutorials (Year 8 – 9)
To conclude...

- Give it a go! Trust the process...
- Seriously ...where does this go?
- The next steps?
  - [http://learnscratch.org/](http://learnscratch.org/) has lesson plans, video tutorials & projects
  - Final word...11 mins in: [https://www.youtube.com/watch?v=xyowJZxrtbg](https://www.youtube.com/watch?v=xyowJZxrtbg)
References

- Scratch: Programming for All

- Creative Computing – a design-based introduction to computational thinking

- Empowering kids to create and share programmable media
  http://web.media.mit.edu/~mres/papers/interactions-scratch-08.pdf

- Creative Coding: Programming for Personal Expression
  http://download.scratch.mit.edu/CreativeCoding.pdf

- Playing and Making Games for Learning
  http://www.gse.upenn.edu/~kafai/print/pdfs/playing.pdf