Explorers need maps: Abstraction, representations and graphs

Paul Curzon
Queen Mary University of London

With support from Google,
D of E, the Mayor of London
and CHI+MED

www.teachinglondoncomputing.org
Twitter: @TeachingLDNComp
Aims

• Give you deeper understanding of core topics
  – Abstraction and data representation in problem solving
  – Graph data structures and finite state machines
  – Computational thinking

• Give you practical ways to teach computing in a fun, thought provoking way
  – away from computers, focus on concepts

• Linked activity sheets and booklets can be downloaded from our website:

  www.teachinglondoncomputing.org
The Knight’s Tour Puzzle
Adapted from an idea by Maciej Syslo & Anna Beata Kwiatkowska, Nicolaus Copernicus University
A Puzzle to solve

• To get you thinking about problem solving let’s start with a puzzle to solve…
• What strategies do you use to try to solve it?
• Is it a hard or easy puzzle?
  • What age group is it suitable for?
The Knight’s Tour Puzzle

• Work out an algorithm for a chess Knight to visit all squares on the board returning to the start
• Record the steps (the algorithm)
How difficult is it?

- What strategies do you use to solve it…
- Is it a simple puzzle?
  - What age group could tackle it?
- Let’s look at a simpler puzzle and return to the Knight’s Tour later
The Tour Guide Puzzle
The Tour Guide Puzzle

- Starting at the hotel, plan a route so that tourists can visit every tourist attraction just once ending up back at the hotel.
- Record the steps (the algorithm)

The Tube Map
How difficult is it?

• What strategies do you use to solve it…
• Is it a simple puzzle?
  • What age group could tackle it?

• Is it simpler than the Knight’s Tour?
  • Why / Why not?
The Tour Guide Solution

- Starting at the hotel, this route visits every tourist attraction just once ending up back at the hotel.
- There are many others possible!

The Tube Map
What is a graph?

- The tube map is in computing terms a ‘graph’ data structure.
- A graph is just a way of representing information about links between things.
- It consists of:
  - nodes (circles) showing ‘places’.
  - Edges (lines) showing which places are linked.
  - A directed graph uses arrows to show one-way links (eg one-way streets).
The Knights Tour Puzzle (again)
The Knight’s Tour as a graph

• Draw the Knights Tour puzzle as a graph.
  • Use nodes for squares
  • Edges show which squares you can jump to from each square

• The graph is the map you draw as you explore the ‘state space’ of the puzzle.
• Once you have a map, answering questions about it is easier
Draw a graph of the Knight’s Tour puzzle

• You need an algorithm to exhaustively explore the state space
  • ie explore possible puzzle positions
An algorithm for drawing graphs

• Start at the start position, draw a node (a circle) and label it with the square number.
• Draw edges (arrows) to each place you can jump to from it, adding new nodes for those places.
• Repeat for each new node added until there are no new nodes added.

• The same algorithm is used to create graphs of gadget interaction design
Here’s a graph of the Knight’s Tour
Now solve the puzzle …
Here’s a solution of the Knight’s Tour puzzle
Is it simpler than the Knight’s Tour?

- In fact they are two versions of exactly the same puzzle
  - Identical if viewed through an abstraction
- Solve one and you’ve solved the other
- The graph abstraction gives a clear map of the problem.
- It throws away spurious information, highlighting the information that matters.
Generalisation

• Solve one and you have solved both
  • once they are generalised to the same graph problem
• We also came up with a general algorithm for creating a graph of a puzzle.
• Finding a circuit round a graph is called a Hamiltonian Circuit problem
  • We could now develop an algorithm for finding a Hamiltonian Circuit of any graph.
Computational Thinking
Lessons

- Algorithmic thinking
- Logical Thinking
- Abstraction
- Generalisation
HexaHexaFlexagon
Automata
Hexahexaflexagons

- Hexahexaflexagons are simple folded paper puzzles.
  - Fold it up and unfold it from the middle to reveal new sides.
- There are 9 ‘sides’ to the flexagon. Find them all.
- Make a map (a graph) to allow you to move at will around the flexagon.
- Use your new found knowledge of graphs to explore the hexahexaflexagon.
- Find a Hamiltonian circuit!
Here’s a graph of the Hexahexaflexagon
Finite State Machines

• A directed graph can actually be thought of as a ‘program’: a finite state machine.
• It describes computation
Finite State Machines

- A finite state machine has
  - Nodes that represent ‘states’ of the machine
  - A start state (one specific node to start from)
  - Edges that represent ‘transitions’ between states
    - They have labels giving the action that will lead to the transition being taken.
  - Outputs: what happens when you are in a state
The Hexahexaflexagon finite state machine

State
Start state
Transition
Action
Output (yellow)
Finite State Machines as computational models

- Use finite state machines to do computational modelling of whatever it describes.
- Actions are inputs that move us from state to state and outputs get printed.
- Used to rapidly prototype devices
- Used to help plan the design of interfaces, websites, the modes of a device, etc
- Eg Now write a Scratch simulation of a flexagon based on the finite state machine
Modelling gadgets, websites, etc

- All this applies to gadgets (and software generally)
- Take your digital watch, central heating controller, digital radio, ...
- Create finite state machines of them
- Then use it as the basis to write your own program
- Also make graphs as a model of a website

Working out the finite state machine of an alarm clock
Checking properties

- Use to check properties of the design
  - is it easy to get back to the home state from any state…
  - Does an action have the same effect everywhere,
  - Can important tasks be done in few steps
  - etc

- We are using this to check the safety of medical devices with regulators
Summary

The way you represent information has a powerful effect on the ease of problem solving

- Graphs are a good representation for any problem that involves links between ‘places’
- Finite state machines turn them in to computational models
- Check designs work
More support

On our website to support this session:
• Activity sheets
• Story sheets
• Slides

Details of more workshops/courses
• free unplugged sessions
• subsidised courses (e.g. on A’level computing)

www.teachinglondoncomputing.org
Twitter: @TeachingLDNComp

• See also www.csunplugged.org for more unplugged finite state machine activities
Together we are
Teaching London Computing

Thank you!

www.teachinglondoncomputing.org
Twitter: @TeachingLDNComp