How can computation... create animated stories?

Flowchart:
- Visual representation of steps of algorithm
- Summarizes how algorithm behaves given specific answers

Symbols
- Boxes: Represent states (or actions or actions)
- Arrows (or edges): Show transitions (or decisions) between states

Flowchart for Animated Story
Animated Story: Behaves the same every time
- No decisions!
- Flowchart is summary of action of story

How to create flowchart?
- Identify Initial State or Scene
- Group individual actions into higher-level “scenes”
  - Somewhat subjective (no right answer)
  - Label with descriptive name
- Identify characters of story
  - Specify actions of each character in scene
- Connect scenes sequentially
See Scratch code posted on web site!

*Initial state*
Background: Railroad
Cat: Left side
Dog: Right side

*Introductions*
Cat: What should we do?
Dog says: I don’t know
Flowchart of Animated Story

*Initial state*
Background: Railroad
Cat: Left side
Dog: Right side

*Introductions*
Cat: What should we do?
Dog says: I don’t know

*Color*
Cat says: Let’s do Tricks!
Cat: I can change colors!
Cat changes to 5 colors
Cat changes back

*Whirl*
Cat says I can whirl
Cat whirls 5 times
Cat changes back

*Fly*
Cat moves 5 times
Cat goes back

*Tiny*
Cat becomes tiny
Dog says I like that!
Cat says why?
How to Transform Flowchart to Scripts?

Approach

• For each scene in flowchart, specify a script
  - Blocks in script show individual actions to be performed
  - Specify script for each character that does something

How to determine when script can run?

• When should “Initial state” run?
  - When GreenFlag is Clicked
• When should dog say “I don’t know” in “Intro”?
  - After cat says “What should we do today?”
• How will dog know cat has finished saying that??
How to Run Desired Script?

Wait for Scripts to Complete!

Beauty of Abstraction

Abstraction: Separation of high-level view of entity from low-level details of implementation

When sender broadcasts “jump”, doesn’t know how “jump” is implemented by different Sprites

Why is this good?
• Simplifies concerns of sender (don’t need to know everything)
• Can change implementation of “jump”
Of course, receiver might not implement “jump”!

Treachery of Images

Ceci n’est pas une pipe.
Naming Convention for Messages

Use good descriptive names
- Purpose of names = help others understand your code
- Suggestion: Name matches name of scene
  - Intro, Color, Whirl, Fly, Tiny, Eat, Fadeout

Problem:
Hard to follow flow of messages across Sprites

Solution:
1. Use Stage to control action as much as possible
2. Use naming convention to help understanding
   - SceneName : Receiver
   - e.g., "Eat : Dog"

Programming Concepts

General
- Divide high-level functionality into logical units
- Descriptive naming is important
- Initial state must be specified
- Incrementally test code as you go
- Scripts must be activated to run
  - When flag clicked; When receive message
- Execution within script proceeds sequentially
- Control : forever, repeat <times>, repeat until
- Parameters (to blocks) specify behavior
- Goal is to make "non-fragile" code

Check-Up

- In your animated story all the scripts are running at the same time. What did you do wrong?
- Can a Sprite receive a message it broadcast?
- When will this code work correctly? when not?
Homework 2: Extra Credit

http://scratch.mit.edu/galleries/view/176626