Lecture 6:
How does computation ... create art
What are different approaches of “Computer Art”?

1. Human uses computer as drawing/painting tool
2. Human designed algorithm computer follows to create exact picture (e.g. drawing in Scratch)
3. Human designed algorithm w/ randomness – human examines results, picks appealing
4. Human designed algorithm w/ randomness – computer evaluates and shows best
5. Human interacts with computer (e.g., according to some algorithm, different sounds create different shapes; different volumes create different sizes)
Art in Scratch: Pen

How to Control Size of Pen?

```
pen down
set pen size to 0
set pen color to light blue
set the pen thickness to 20
move
```

How to Control Shade of Pen?

```
start leaving a pen trail
set the pen color to light blue
set the pen thickness to 20
move
```

Art in Scratch: Stamp

```
clear all the stamps and lines
repeat 0 times:
move
turn
stamp a print of your costume on the stage
```

Progression of Art Programs in Today's Lecture

1. Paint tool
2. Random movements, random color
3. Random turns, random color
4. Progressively larger turns: Interact with user for color
5. Complete user control
Example 1: Paint Program

Program (Problem) Specification

- Describes precisely how computation behaves
  - How should inputs be transformed into outputs?
- What is Input?
  - User typing on keyboard
  - Mouse actions
  - Messages arriving over network
  - Data read from files
  - Any other sensors (GPS location, motion)
- What is Output?
  - Anything sent to display
  - Anything sent to printer
  - Messages sent over network
  - Data stored permanently in files

Example 1: Painting Program

Specification

Mouse:
- Use to move Paintbrush
- When click, draw

Key 'C':
- Change color of paintbrush

Up and Down arrows:
- Change size of paintbrush

Space bar:
- Clears Stage

Example 2: Brownian (Random) Motion

Specification

Stage is empty
- Pen begins in middle of stage

Repeat until reach edge
- Move randomly up/down and left/right
- Change to random (nearby) color
Example 2: Brownian (Random) Motion

Stage is empty
Pen begins in middle of stage
Repeat until reach edge
  • Move randomly up/down and left/right
  • Change to random (nearby) color

Example 3: Random Turns

Stage is empty
Pen begins in middle
Repeat until reach edge
  • Move in irregular arc of circle
  • Change to random (nearby) color

Example 4: Increasing Turns

Stage is empty
Begin w/ 0 degree turn
Forever:
  • Turn one degree more on each stamp
  • When reach 45 degrees, set back to 0
  • Change color random amount (controllable by user)
**Variables: Remember Values**

- Variables can hold different values over time.
- Assignment statements ("set variable to" in Scratch) are not mathematical equations.
- New value is calculated and placed in variable.

**Variables**

- Hold different values over time.
- Example:
  - Stage is empty
  - Forever:
    - Turn one degree more each time
    - When reach 45 degrees, set back to 0
    - Change color random amount

**Example 4: Increasing Turns**

- Variables can hold numbers or strings.
- Can be used anywhere a number or string is used.
Example 5: Complete Control
Pen constantly moving and turning
Up/down arrows: Change step size
Left/right arrows: Change Turn amount
C: Change color
S/D: Change size
Space: Clear screen

Programming Concepts
General
- Divide high-level functionality into logical units
- Descriptive naming is important
- Initial state must be specified
- Goal is to make "non-fragile" code
- Incrementally test code as you go
Data Structures
- Decision trees: states plus transitions
Variables
- Parameters (to blocks) specify behavior
  - Creating new variables

Scratch Features
Execution within script proceeds sequentially
Basic blocks
- Tour of many Motion and Looks blocks
- X-Y coordinate system for Stage
- Pen and stamps
- Random numbers
Activating scripts
- When Green Flag is Clicked (initial state)
  - When I receive <message>: Corresponding block: broadcast <message>
Data Types
- Ask questions: User types string "answer"
- String manipulation: letter <x> of <string>
  - Variables: Numbers integers
Control Blocks
- Forever
- repeat <number of times>
- repeat until <condition is true>
- if <condition> then <action1> else <action2>

Today’s Summary
Today’s Topics
- Can create art with pen and stamp tools
- Variables: Allow program to “remember” information
  - Value held in variable can change (vary) as program runs
Reading: TED Talk
- Golan Levin makes art that looks back at you
Announcements
- Exercises on Slides posted from Lecture 5
- Homework 2 due before class Wednesday
  - Create animated story
- Any questions: Contact me or TA – We are here to help