Introduction to Scratch and Algorithms

BU Summer Challenge
Anton Njavro
Agenda for Today

- Introduction to programming concepts
- Algorithms
- Scratch language
Why Scratch?

- Created at MIT, Scratch is an easy and fun to use language allowing us to explore basic programming concepts with ease.
- Allows for branching, control sequences, number crunching and many other things.
- Easy to incorporate animations for a more visual feel of programming.
Important Blocks of Programming

- Function
- Input
- Output
- Branching
- Looping
- Interrupts
Let's Create Space Invaders
Group Project - Flappy Bird
Break
Algorithms

- An algorithm is a finite sequence of well-defined instructions, typically used to solve a class of problems or perform a computation.
- Algorithms are fundamental to computer science as they provide a step-by-step method for solving problems and performing tasks efficiently.
Characteristics of an Algorithm

- Finiteness
- Definiteness
- Input
- Output
- Effectiveness
Types of Algorithms

- **Brute force**
  - Straightforward approach, typically tries all possible solutions.

- **Divide and conquer**
  - Break problem into smaller subparts, solve smaller parts and combine solutions together. Often used in relation to **recursion**.

- **Greedy Algorithms**
  - Make locally optimal choice at each step.

- **Backtracking**
  - Try all possible solutions and eliminate those failing to meet the criteria.
Analysis of Algorithms

- Time complexity
  - How long is the runtime of an algorithm relative to its input.

- Space complexity
  - Measures memory usage of an algorithm relative to its input.

- Big O notation
  - Mathematical notation used to describe the upper bounds on time and space complexity.
Big O Notation

- $O(1)$: Constant time, runtime doesn’t change with input size.
  - Accessing element in an array
- $O(\log n)$: Logarithmic complexity
  - Binary search
- $O(n)$: Linear
  - Linear search
- $O(n \log n)$
- $O(n^2)$
- $O(2^n)$
Sorting Examples

- Bubble Sort in 2
- Merge Sort in 3
**Divide and Conquer**

- A paradigm that breaks a problem into smaller sub-problems, solves each sub-problem recursively, and combines their solutions.
- **Example: Merge Sort**
  - Divide: Split the array into two halves.
  - Conquer: Recursively sort each half.
  - Combine: Merge the two sorted halves into a single sorted array.
Greedy Algorithm

- A paradigm that makes the locally optimal choice at each step with the hope of finding the global optimum.
- Example: Dijkstra’s Algorithm:
  - Used for finding the shortest path from a source node to all other nodes in a weighted graph.
  - Steps:
    - Initialize distances
    - Select the nearest unvisited node
    - Update distances
    - Repeat until all nodes are visited.
Group Project - Racing Game

"Cocoa River"
by Ozoneone