Introduction to Programming Using Scratch

Brooklyn College

M. Meyer
Scratch is an IDE (Integrated Development Environment) that allows users to create and run simple graphics programs.

Users create programs in Scratch using an imperative, procedural, object-oriented programming language that has very simple syntax*.

* Don't worry, all of these terms will be made clear to you.
Programming Languages:

1. Allow us to "talk" to a computer, in a language that we can understand.
2. Require a well-defined syntax.
3. Allow us reuse and share code (functions and libraries).
4. Fall into a variety of different paradigms: Functional, Logical, Imperative, Procedural and Object-Oriented.

(Scripting languages are a sub-type)
Simple Development

1. Scratch requires very little typing in order to create programs.
2. Visual code creation using drag and drop.
Simplified Syntax

1. No missing semi-colon problems.
2. Code blocks can only fit together in pre-defined way.
Large library.

1. Over 100 predefined functions and limited ability to make more.
2. Functions cover vast majority of things your objects to be able to do in a graphics program.
Object-Oriented (1)

1. OO programming is an extremely important programming paradigm.
2. Scratch is not true OO programming, but good example of basic concepts:
   - Creating programs that are composed of interacting objects.
   - These objects have associated properties and functions.
Object-Oriented (2)

1. Objects in Scratch are called "Sprites".
2. Properties of Sprites include:
   - Location
   - Look
   - User defined properties (variables).
3. Functions of Sprites include:
   - Move
   - Make Sound
   - Detect Collision
**Sprites**

Found in the lower right corner of the screen.

Click on them to select them and change contents of main window.
Imperative Programming

1. The imperative paradigm views a program as a "list" of things to do.

2. Imperative languages need 3 things:
   • **Sequence** -> A predefined order in which to process information.
   • **Selection** -> The ability to make a choice. The "IF" statement.
   • **Repetition** -> The ability to repeat an action. The "WHILE" statement.
Sequence

- All "scripts" processed from top down.
- 4 possible start conditions, 3 end.
Selection

• If, If-else and wait_until functions.
Repetition

- Variety of functions including `repeat_until`.

![Diagram of a block-based programming environment showing a `repeat_until` block with conditions and actions related to `x` position and direction.](image)
Procedural Programming

• The procedural programming paradigm is based upon the concept of the “procedure call”: the ability to “send a message” to another section of a program.

• Procedural programming allows us to create sections of code that can be reused over and over.
This example is trivial. But you could easily have a procedure with a much more complicated function. And coding it this way would mean you only have to write the procedure once. Then call it when you need it.
Benefits

By developing a program in Scratch:

• Experience in OO paradigm.
• Experience in Imperative paradigm.
• Experience in Procedural paradigm.
• Experience in program logic.
• Experience in development lifecycle.
• Experience in debugging.

Plus you have a chance to make something that belongs to you.
Are you an Artist?

Using Sketch you can develop simple animations and visual storyboards. What's your story? Who are the characters? What's the conflict? What do you want your audience to feel? What objects would you need to create? What would their properties/functions be?
Artists

Checkout:

/Projects/Animation
DayDream

/Projects/Stories/
Stargate

/Projects/SpeakUp
AgainstSpammers
Are you a Gamer?

Using Sketch you can develop 2D and "forced 3D" games.
What's your game idea?
What 'genre' will it look like?
How will a player interact with the game?
What do you want the player to feel?
What objects would you need to create?
What would their properties/functions be?
Gamers

Checkout:

/Projects/Games
BugOnAPlate
Tetris

/Projects/Lists/
FruitCraftRPG
RepeatAfterMe