

HƯỚNG DẪN GIẢNG DẠY HỌC SINH

LẬP TRÌNH ỨNG DỤNG BẰNG NGÔN NGỮ SCATCH

• MỤC TIÊU

Khóa học sẽ giới thiệu những kiến thức cơ bản về ngôn ngữ lập trình Scratch. Học sinh sẽ được hướng dẫn để viết những chương trình bằng cách “kéo thả” những khối chức năng có sẵn đơn giản. Các bài thực hành sẽ giúp người học nắm vững các kỹ năng cần thiết cho việc lập trình bằng ngôn ngữ Scratch. Kết thúc khóa học học sinh sẽ có cơ hội dùng ngôn ngữ lập trình này để làm ra một kịch bản phim hoạt hình, một trò chơi đơn giản, hay một bài toán đơn giản.

• PHÂN BỐ THỜI LƯỢNG BẮT BUỘC (1 tiết ~ 45 phút)

- **Tiết 1:** Scratch – Ngôn ngữ lập trình “Kéo Thả”
- **Tiết 2:** Các thao tác căn bản trên câu lệnh Scratch
- **Tiết 3:** Vẽ hình trên Scratch
- **Tiết 4:** Vòng lặp trong Scratch
- **Tiết 5:** Lập trình Hoạt hình trong Scratch
- **Tiết 6:** Lập trình Hoạt hình trong Scratch (TT)
- **Tiết 7:** Lập trình Xuất – Nhập trong Scratch
- **Tiết 8:** Câu lệnh điều kiện trong Scratch

1. Nội dung tóm tắt môn học

- Giới thiệu về ngôn ngữ Scratch
- Các ứng dụng được hiện thực bằng ngôn ngữ Scratch

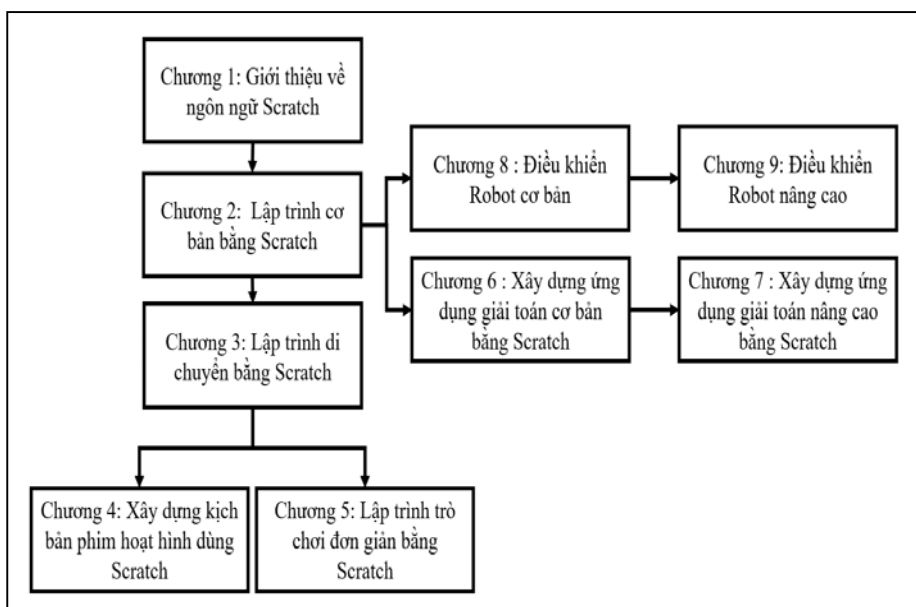
2. Tài liệu học tập

Sách, Giáo trình và tài liệu tham khảo chính:

- [1] Slide bài giảng
- [2] Sách giáo trình
- [3] <https://scratch.mit.edu/>

3. Cấu trúc chương trình

Chương trình gồm 9 chương được phân bố theo cấu trúc như sau:



Dựa vào cấu trúc chương trình trên, nhà trường có thể chủ động lựa chọn các hướng phù hợp để áp dụng ngôn ngữ Scratch vào dạy học, Ví dụ:

- Hướng về lập trình giải toán: Chương 1, 2, 6 và 7
 - Hướng về phim hoạt hình, chuyển động:
 - + Chương 1, 2, 3, 4 hoặc
 - + Chương 1, 2, 3, 5
 - Hướng về Robot: Chương 1, 2, 8 và 9
- Đối với mỗi hướng, chương cuối cùng là chương nâng cao.

4. Tiêu chuẩn đầu ra bắt buộc – học sinh học chương 1,2

Các trường tiến hành dạy học sinh với nội dung ở các chương 1,2. Tổng thời lượng tối thiểu là 8 tiết học. Kết thúc khóa học, học sinh được kiểm tra kiến thức bằng các bài trắc nghiệm. Các học sinh sau khi được học kiến thức của chương 1 và 2 về ngôn ngữ lập trình phải đạt các tiêu chí được liệt kê như bảng dưới đây:

1	Học sinh có khả năng thao tác cơ bản trên môi trường lập trình Scratch
1.1	Học sinh nhận biết chức năng các lệnh lập trình Scratch
1.2	Học sinh có khả năng tự tìm hiểu chức năng các lệnh mới trên Scratch
1.3	Học sinh có khả năng đọc hiểu một chương trình Scratch
1.4	Học sinh có khả năng hiện thực một chương trình Scratch đơn giản

5. Các tiêu chuẩn đầu ra nâng cao trên Scratch như sau:

1	Học sinh có khả năng xây dựng các trò chơi kết hợp với hình ảnh động
1.1	Học sinh có khả năng tự phát triển một trò chơi
1.2	Học sinh có khả năng xây dựng một trò chơi với các câu hỏi kết hợp với các hoạt cảnh hoạt hình
2	Học sinh có khả năng xây dựng các ứng dụng giải toán phức tạp
2.1	Học sinh có khả năng phân tích các bài toán phức tạp có khả năng ứng dụng trên Scratch
2.2	Học sinh có khả năng hiện thực các bài toán phức tạp trên Scratch
3	Học sinh có khả năng sử dụng ngôn ngữ Scratch để điều khiển Robot
3.1	Học sinh hiểu chức năng các lệnh Scratch cho Robot
3.2	Học sinh có khả năng lập trình ứng dụng điều khiển Robot

6. Nội dung chi tiết

Tiết	Nội dung	Phương pháp đánh giá
(4 tiết, bắt buộc)	<ul style="list-style-type: none">▪ Chương 1: Giới thiệu về ngôn ngữ Scratch▫ Ngôn ngữ lập trình là gì▫ Giới thiệu ngôn ngữ Scratch▫ Cài đặt chương trình lập trình▫ Giới thiệu cách kéo thả▫ Giới thiệu cách tổ chức các lệnh trong Scratch▫ Thử các chức năng của các lệnh	<ul style="list-style-type: none">▪ Yêu cầu học sinh kéo thả một số lệnh Scratch ra màn hình và xác định chức năng của lệnh đó.▪ Cho một số chức năng đơn giản, yêu cầu học sinh tìm lệnh tương ứng với các chức năng đó.
(4 tiết, bắt buộc)	<ul style="list-style-type: none">▪ Chương 2: Lập trình cơ bản bằng Scratch▫ Các lệnh về chuyển động▫ Ghép nối các lệnh▫ Tạo bảo sao, xóa lệnh▫ Chỉnh lại vị trí cho nhân vật▫ Lưu – Mở chương trình	<ul style="list-style-type: none">▪ Cho học sinh xem một chương trình có sẵn trên giấy và yêu cầu hiện thực lại▪ Yêu cầu học sinh hiện thực một chuỗi các chuyển động đơn giản, ví dụ: Di chuyển 10 bước, rẽ trái, di chuyển thêm 20 bước
(4 tiết, tự chọn)	<ul style="list-style-type: none">▪ Chương 3: Lập trình di chuyển bằng Scratch▫ Các lệnh về chuyển động▫ Nguyên lý về hiệu ứng chuyển động▫ Giải thích nguyên lý hoạt động của máy tính (vòng lặp)▫ Hiện thực chuyển động cho chú mèo với 2 trạng thái	<ul style="list-style-type: none">▪ Cho học sinh xem một chương trình di chuyển (có hiệu ứng) có sẵn trên giấy và yêu cầu học sinh hiện thực lại trên Scratch▪ Yêu cầu học sinh hiện thực một chương trình di chuyển có hiệu ứng đơn giản
(4 tiết, tự chọn)	<ul style="list-style-type: none">▪ Chương 4: Xây dựng kịch bản phim hoạt hình dùng Scratch▫ Tạo cảnh nền cho kịch bản▫ Thêm nhiều ảnh nền vào chương trình	<ul style="list-style-type: none">▪ Yêu cầu học sinh hiện thực một hoạt cảnh có chú mèo di chuyển trong 2 ảnh nền khác nhau▪ Yêu cầu học sinh thêm các hiệu ứng về âm thanh khi chú

Tiết	Nội dung	Phương pháp đánh giá
	<ul style="list-style-type: none"> ▫ Cho nhân vật di chuyển trong nhiều ảnh nền khác nhau ▫ Thêm các hiệu ứng về âm thanh khi thay đổi ảnh nền 	<p>mèo chuyển qua hoạt cảnh mới</p>
(4 tiết, tự chọn)	<ul style="list-style-type: none"> ▪ Chương 5: Lập trình trò chơi đơn giản bằng Scratch ▫ Xây dựng kịch bản trò chơi ▫ Xây dựng các đối tượng trong trò chơi ▫ Xây dựng quy luật chơi ▫ Hiện thực trò chơi 	<ul style="list-style-type: none"> ▪ Yêu cầu học sinh thiết kế trên giấy ý tưởng và xác định quy luật của một trò chơi ▪ Yêu cầu học sinh xác định rõ bao nhiêu đối tượng trong trò chơi ▪ Học sinh hiện thực trò chơi với sự giúp đỡ của giáo viên (gợi ý về các câu lệnh nên được sử dụng)
(4 tiết, tự chọn)	<ul style="list-style-type: none"> ▪ Chương 6 : Xây dựng ứng dụng giải toán cơ bản bằng Scratch ▫ Đặt yêu cầu bài toán ▫ Xây dựng phương pháp giải ▫ Xây dựng chi tiết các bước giải ▫ Xem xét các trường hợp ngoại lệ ▫ Hiện thực bằng Scratch 	<ul style="list-style-type: none"> ▪ Yêu cầu học sinh trình bày ra giấy phương pháp giải một bài toán, ví dụ giải phương trình bậc nhất, tính diện tích hình tam giác với độ dài 3 cạnh ▪ Học sinh hiện thực chương trình giải toán với sự giúp đỡ của giáo viên (giáo viên góp ý những trường hợp ngoại lệ, những câu lệnh nên được sử dụng)
(4 tiết, tự chọn)	<ul style="list-style-type: none"> ▪ Chương 7 : Xây dựng ứng dụng giải toán nâng cao bằng Scratch ▫ Giới thiệu các bài toán nâng cao như giải hệ phương trình nhiều ẩn số hay các bài toán về tính giai thừa ▫ Xây dựng phương pháp giải ▫ Xây dựng chi tiết các bước giải ▫ Xem xét các trường hợp ngoại lệ 	<ul style="list-style-type: none"> ▪ Yêu cầu học sinh trình bày ra giấy phương pháp giải một bài toán, ví dụ giải hệ phương trình 3 ẩn số ▪ Học sinh hiện thực chương trình giải toán với sự giúp đỡ của giáo viên (giáo viên góp ý những trường hợp ngoại lệ, những câu lệnh nên được sử dụng)

Tiết	Nội dung	Phương pháp đánh giá
	<ul style="list-style-type: none"> ▫ Hiện thực bằng Scratch 	
(4 tiết, tự chọn)	<ul style="list-style-type: none"> ▪ Chương 7 : Điều khiển Robot cơ bản ▫ Tìm hiểu môi trường lập trình Scratch cho Robot ▫ Giới thiệu cấu tạo của Robot ▫ Nguyên lý và lập trình chớp tắt bóng đèn trên Robot ▫ Nguyên lý và lập trình điều khiển động cơ trên robot 	<ul style="list-style-type: none"> ▪ Yêu cầu học sinh lập trình điều khiển robot với các yêu cầu sau: ▫ Robot chạy tới thì sáng cả 2 bóng đèn ▫ Robot chạy lùi thì tắt cả 2 bóng đèn ▫ Robot rẽ trái thì đèn bên trái sáng, bên phải tắt ▫ Robot rẽ phải thì đèn bên phải sáng, bên trái tắt
(4 tiết, tự chọn)	<ul style="list-style-type: none"> ▪ Chương 8: Điều khiển Robot nâng cao ▫ Giới thiệu các cơ chế chạy tự động cho Robot ▫ Hiện thực cơ chế chạy tự động cho Robot 	<ul style="list-style-type: none"> ▪ Yêu cầu học sinh hiện thực một cơ chế chạy tự động như sau: ▫ Khi nhấn nút mũi tên tới, robot chạy tới ▫ Khi nhấn nút mũi tên lùi, robot dừng hẳn ▫ Trong quá trình chạy, nhấn nút rẽ phải, robot dịch đầu qua phải rồi tự động chạy tới ▫ Trong quá trình chạy, nhấn nút rẽ trái, robot dịch đầu qua trái rồi tự động chạy tới.

7. Phương pháp kiểm tra đánh giá học sinh

Kết thúc 8 tiết học về Ngôn ngữ Lập trình Scratch, học sinh sẽ có 2 bài kiểm tra.

- Làm bài tập kiểm tra theo nhóm với các nội dung được gợi ý qua từng tiết học, lưu và nộp bài cho giáo viên dưới dạng tệp (file). Các trường tổng hợp kết quả của học sinh và gửi về Quỹ Dariu theo địa chỉ: pri.ho01@dariu.org – *Trần Thanh Hải*.
 - Mỗi nhóm tối đa 10 em học sinh.
 - Hình thức làm bài tập nhóm: Yêu cầu học sinh hoàn thiện được một chương trình Scratch hoàn thiện ở mức đơn giản nhất: Vẽ được một hình, thực hiện lại một ví dụ đã sửa ở trên lớp...
 - Giáo viên các trường chấm điểm nhóm các em học sinh theo hai mức: Đạt – Không đạt.
 - ✓ **Yêu cầu đối với một chương trình của học sinh đạt như sau:**
 - + Các em học sinh có thể mở và lưu trữ một chương trình
 - + Trong chương trình của các em học sinh có tối thiểu 10 câu lệnh
 - + Chương trình của các em học sinh hoàn thiện ở mức cơ bản nhất, có thể chạy được
 - ✓ **Nhóm học sinh không đáp ứng được yêu cầu trên, sẽ xếp loại không đạt.**
- Làm bài kiểm tra online trên hệ thống kiểm tra của Quỹ Dariu.
 - Số lượng câu hỏi: 10 câu; Thời gian làm bài: 20 phút.

TIẾT 1:

SCRATCH – NGÔN NGỮ LẬP TRÌNH “KÉO THẢ”

• MỤC TIÊU

- Học sinh hiểu được khái niệm ngôn ngữ lập trình là gì
- Học sinh nắm được tư tưởng lập trình “kéo thả”
- Học sinh làm quen với giao diện lập trình trên Scratch
- Giáo viên và học sinh cài đặt công cụ lập trình trên máy tính của trường để giảng dạy những tiết học sau.

• PHÂN BỐ THỜI LƯỢNG

1. Ngôn ngữ là gì?
(5 phút)
2. Ngôn ngữ lập trình là gì?
(5 phút)
3. Ngôn ngữ lập trình Scratch
(5 phút)
4. Giao diện lập trình trên Scratch
(15 phút)
5. Cài đặt công cụ lập trình cho Scratch
(15 phút)

1. Ngôn ngữ là gì? (5 phút)

Để tìm hiểu khái niệm về ngôn ngữ lập trình, trước tiên chúng ta khái quát qua một số đặc điểm đặc trưng của khái niệm ngôn ngữ. Ngôn ngữ là một **hệ thống** để con người có thể **giao tiếp và liên lạc** với nhau. Nói một cách khác, ngôn ngữ là công cụ để con người có thể truyền đạt những tư tưởng của mình cho người khác. Tuy nhiên, khi hai người không dùng một hệ thống ngôn ngữ, thì vấn đề giao tiếp sẽ được thực hiện như thế nào?



Hình 1: Hai hệ thống ngôn ngữ giao tiếp với nhau thông qua "Thông dịch viên"

Hình 1 là một ví dụ về hai hệ thống ngôn ngữ khác nhau: Tiếng Việt và Tiếng Anh. Trong thực tế, thông thường một thông dịch viên sẽ đóng vai trò là cầu nối để liên kết giữa hai hệ thống ngôn ngữ. Và hiển nhiên, thông dịch viên phải có khả năng hiểu được hai ngôn ngữ.

2. Ngôn ngữ lập trình là gì? (5 phút)

Tương tự như khái niệm ngôn ngữ bên trên, ngôn ngữ lập trình là hệ thống ngôn ngữ dành cho máy tính, giúp máy tính hiểu và thực hiện một số chức năng nào đó. Không may mắn cho chúng ta, ngôn ngữ dành cho máy tính rất phức tạp vì nó chỉ làm việc với các kí hiệu 0 và 1. Ngôn ngữ dành cho máy tính còn có một tên gọi khác, là ngôn ngữ máy.



Hình 2: Ngôn ngữ tự nhiên và ngôn ngữ máy

Ở thời kì đầu, khi máy tính mới được phát minh, chỉ những chuyên gia mới có thể làm việc được với máy tính. Tuy nhiên, qua thời gian, con người bắt đầu thiết kế những “thông dịch viên máy tính” để có thể kéo gần khoảng cách giữa ngôn ngữ tự nhiên và ngôn ngữ máy. Với sự trợ giúp của thông dịch viên, một hệ thống ngôn ngữ mới được đưa ra, và được gọi là ngôn ngữ lập trình. Ngôn ngữ lập trình thế hệ mới giúp cho việc tương tác giữa con người với máy tính được gần gũi hơn và thuận tiện hơn.

3. Ngôn ngữ lập trình Scratch (5 phút)

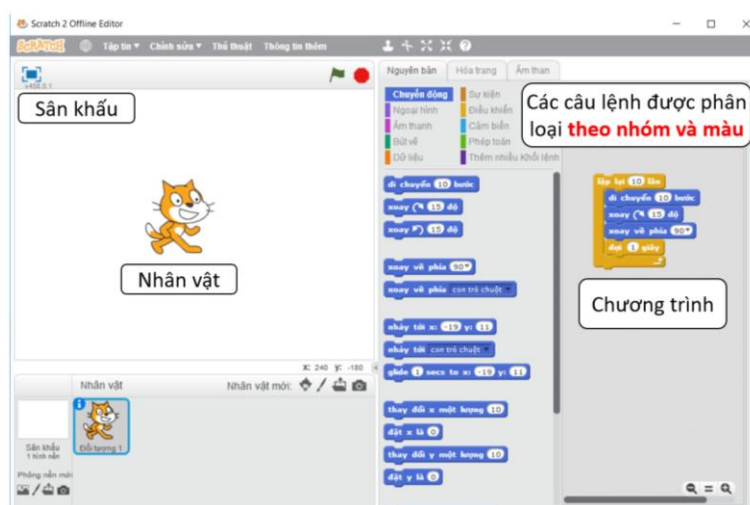
Ngôn ngữ Scratch được ra đời theo xu thế phát triển của ngôn ngữ lập trình, với mục tiêu đơn giản hóa việc lập trình của con người và hướng tới số đông người dùng. Sử dụng Scratch, người dùng không còn phải gõ từng câu lệnh như các ngôn ngữ truyền thống (ví dụ Pascal). Việc viết một chương trình được thực hiện bằng cách “kéo thả” các câu lệnh tương ứng và ráp nối chúng lại với nhau. Môi trường lập trình trên Scratch vì vậy, đặc biệt thích hợp cho mọi lứa tuổi, mọi ngành nghề và không đòi hỏi nhiều kĩ năng lập trình khi mới bắt đầu.

Một số thông tin bên lề về Scratch, chúng ta có thể tìm hiểu thêm trên mạng ở địa chỉ <https://scratch.mit.edu>. Đây là hệ thống ngôn ngữ lập trình do nhóm nghiên cứu Lifelong Kindergarten Group thuộc đại học MIT (Massachusetts Institute of Technology) thiết lập đầu năm 2008. Ý tưởng ban đầu của nhóm chỉ là thiết lập một ngôn ngữ lập trình mới, đơn giản, chỉ dùng kéo thả, dành cho trẻ con để thiết lập trò chơi, phim hoạt hình, ứng dụng đơn giản, kích thích sự sáng tạo trong môi trường làm việc nhóm của trẻ.

Tuy nhiên Scratch chỉ thực sự bùng nổ từ năm 2014 khi một số quốc gia như Anh, Mỹ đã đổi mới đột phá chương trình giảng dạy môn Tin học trong nhà trường, đưa nội dung kiến thức Khoa học máy tính vào nhà trường ngay từ cấp Tiểu học. Việc điều chỉnh chương trình môn Tin học này đã kéo theo sự gia tăng bùng nổ của Scratch trên phạm vi toàn thế giới. Số lượng học sinh đăng ký tham gia trang Scratch tăng đột biến cả về số lượng và chất lượng. Thực tế đã chứng minh tính hấp dẫn của các môi trường lập trình kéo thả như Scratch, sự đam mê lập trình của trẻ nhỏ. Scratch vô cùng thích hợp cho trẻ lứa tuổi từ 6 đến 14, tức là các cấp Tiểu học, Trung học cơ sở ở Việt Nam. Ngoài ra, Scratch cũng rất thích hợp để tạo ra các ứng dụng đồ họa, animation, bài học, bài giảng, mô phỏng kiến thức, trình diễn, sách điện tử, trò chơi, ... rất phù hợp với nhà trường, giáo viên, học sinh.

4. Giao diện lập trình trên Scratch (15 phút)

Hình 3 bên dưới là giao diện và các đối tượng khi làm việc với ngôn ngữ lập trình scratch



Hình 3: Giao diện lập trình ngôn ngữ Scratch

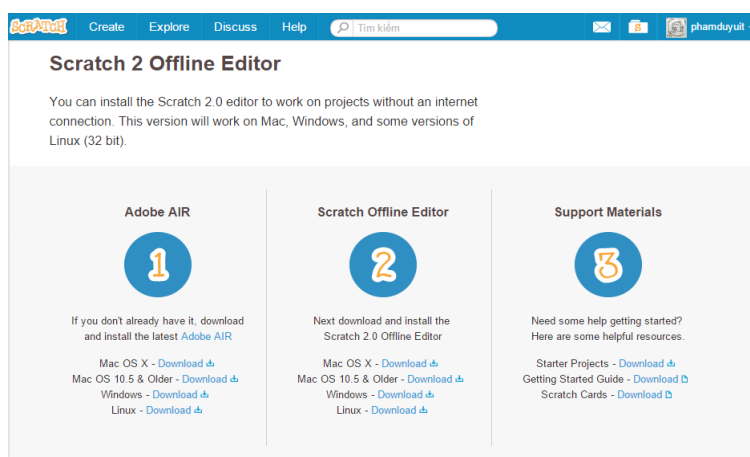
Nhân vật và Sân khấu: Thay vì phải lập trình bằng các câu lệnh và in kết quả ra màn hình, với ngôn ngữ Scratch, các câu lệnh sẽ tạo ra các hiệu ứng khác nhau trên nhân vật của chúng ta, mặc định là chú mèo Scratch. Chú mèo có thể di chuyển, phát ra âm thanh cũng như làm được rất nhiều hoạt cảnh thú vị trên một sân khấu.

Các câu lệnh: Các câu lệnh trong Scratch sẽ được phân loại theo từng nhóm, mỗi nhóm có một màu sắc riêng, giúp chúng ta có thể dễ dàng nhận dạng cũng như tìm kiếm. Scratch hỗ trợ khá nhiều các câu lệnh có sẵn, khoảng trên dưới 100 câu lệnh. Do vậy, việc tìm kiếm chúng dựa vào màu sắc sẽ tiết kiệm được rất nhiều thời gian. Ví dụ khi chúng ta thấy một câu lệnh có màu xanh dương, thì chúng sẽ nằm trong nhóm các câu lệnh về chuyển động.

Chương trình: Đây là nơi chúng ta “kéo thả” và ráp nối các câu lệnh lại với nhau, để tạo thành một chương trình điều khiển chú mèo Scratch. Tổng quát, chương trình là một tập hợp gồm nhiều câu lệnh.

5. Cài đặt công cụ lập trình cho Scratch (15 phút)

Để cài đặt công cụ lập trình trên Scratch, chúng ta truy cập vào đường dẫn trực tuyến sau đây: <https://scratch.mit.edu/scratch2download>. Trang thông tin sau đây sẽ hiện ra:



Hình 4: Trang chủ tải các phần mềm cần thiết để lập trình Scratch

Bước 1: Chúng ta cần tải và cài đặt Adobe AIR (phiên bản cho máy sử dụng Windows)



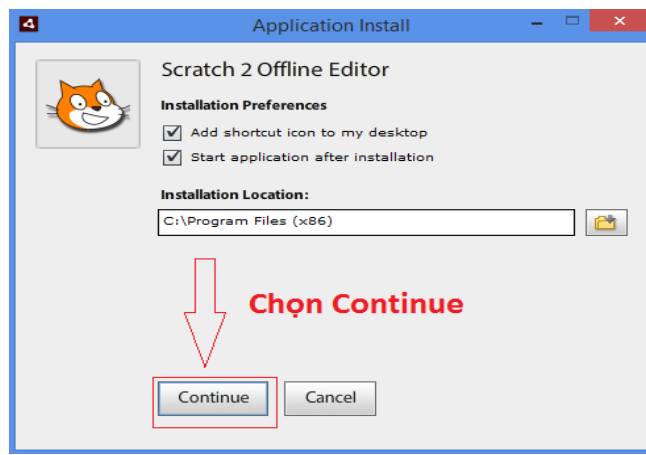
Hình 5: Tải và cài đặt Adobe AIR

Bước 2: Tải và cài đặt Scratch



Hình 6: Tải và cài đặt Scratch

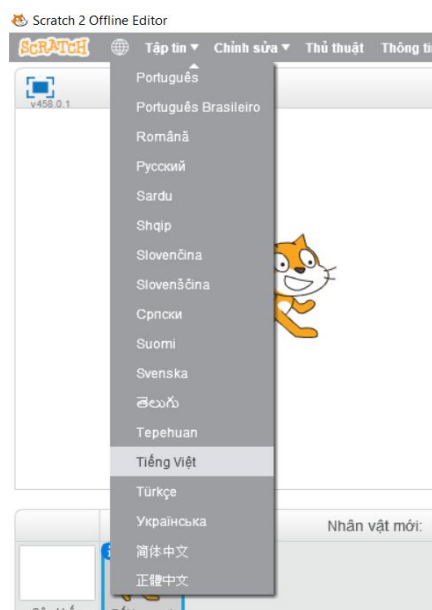
Bước 3: Nhấp đôi vào file tải về, màn hình sau đây sẽ hiện ra:



Hình 7: Bắt đầu cài đặt Scratch

Chúng ta chọn Continue, và chọn YES nếu có thông báo hiện ra và chờ cho đến khi cài đặt hoàn tất, cửa sổ giao diện Scratch như Hình 3 sẽ hiện ra.

Bước 4: Chọn vào biểu tượng quả địa cầu và cài đặt Tiếng Việt cho chương trình



Hình 8: Cài đặt Tiếng Việt cho chương trình.

TIẾT 2:

CÁC THAO TÁC CĂN BẢN TRÊN CÂU LỆNH SCRATCH

• MỤC TIÊU

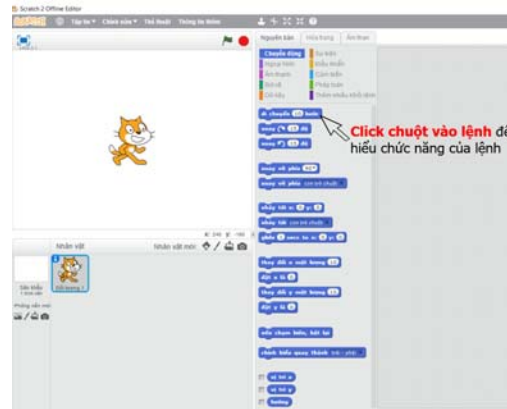
- Học sinh nắm được cách tự tìm hiểu chức năng của một câu lệnh.
- Học sinh nắm được cách tương tác với các câu lệnh trong Scratch bao gồm:
 - + Kéo thả một câu lệnh
 - + Hiệu chỉnh một câu lệnh
 - + Tạo bản sao một câu lệnh
 - + Xóa một câu lệnh
 - + Ghép các câu lệnh
 - + Tách các câu lệnh
- Học sinh có khả năng tự hiện thực một chương trình có sẵn.

• PHÂN BỐ THỜI LƯỢNG

1. Tìm hiểu chức năng một câu lệnh (5 phút)
2. Kéo thả câu lệnh vào trong chương trình (5 phút)
3. Thay đổi thông số cho câu lệnh (5 phút)
4. Tạo bản sao cho một câu lệnh (5 phút)
5. Xóa một câu lệnh (5 phút)
6. Ghép nối các câu lệnh (5 phút)
7. Tách các câu lệnh (5 phút)
8. Bài tập trên lớp (10 phút)
9. Bài tập về nhà

1. Tìm hiểu chức năng một câu lệnh (5 phút)

Môi trường lập trình trên Scratch cho phép người dùng có thể tự tìm hiểu chức năng của các câu lệnh một cách rất nhanh chóng và thuận tiện. Để có thể tự học tác dụng của một câu lệnh, chúng ta chỉ việc nhấp chuột trái vào nó, khi đó chú mèo Scratch sẽ phản ứng lại với câu lệnh. Hình 1 là ví dụ khi chúng ta nhấp vào lệnh di chuyển 10 bước. Câu lệnh này có màu xanh dương và thuộc nhóm **Chuyển động**



Hình 1: Nhấp chuột vào một lệnh để tìm hiểu chức năng của nó

Thực hành trên máy tính: Giáo viên cho học sinh tìm hiểu chức năng của 4 câu lệnh sau đây và yêu cầu học sinh phát biểu chức năng của các câu lệnh này, dựa vào phản ứng của nhân vật khi nhấp chuột vào từng câu lệnh này.

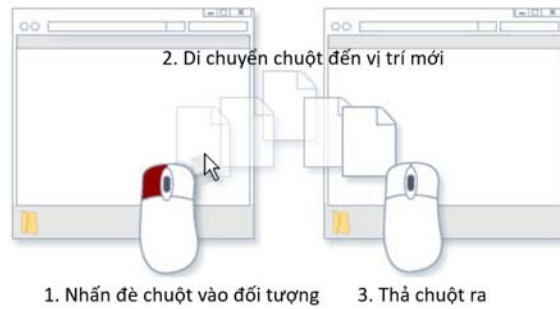
Câu lệnh	Chức năng
	Nhân vật sẽ di chuyển về phía trước 1 đoạn ngắn
	Nhân vật sẽ xoay về bên phải 15 độ, tính từ vị trí hiện tại
	Nhân vật sẽ xoay về bên trái 15 độ, tính từ vị trí hiện tại
	Nhân vật sẽ hướng về bên phải, bất chấp vị trí hiện tại đang ở đâu

Bảng 1: Một số câu lệnh cơ bản trên Scratch

Trong các câu lệnh trên, chúng ta chú ý câu lệnh cuối cùng: Dù nhân vật đang xoay về phía nào, khi nhấn vào câu lệnh này, nhân vật của chúng ta luôn hướng về phía bên phải. Đây sẽ là câu lệnh để chúng ta đặt lại vị trí cho nhân vật về trạng thái ban đầu.

2. Kéo thả câu lệnh vào trong chương trình (5 phút)

Đây là tác vụ căn bản khi làm việc với Scratch. Khi viết một chương trình, chúng ta chỉ việc “kéo-thả” các câu lệnh tương ứng. Giáo viên nhắc lại cho học sinh tác vụ “kéo-thả” sử dụng chuột như minh họa ở Hình 2.

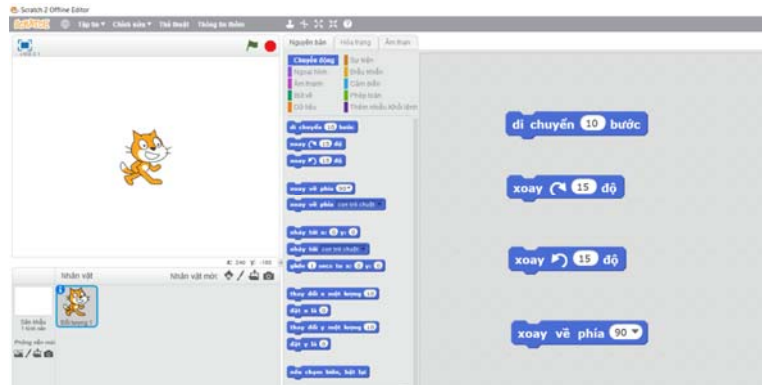


Hình 2: Ba bước để thực hiện tác vụ "kéo-thả"

Tác vụ kéo thả có 3 bước như sau:

1. Nhấn đê chuột trái vào đối tượng
2. Di chuyển chuột cùng với đối tượng đến vị trí mới
3. Thả chuột trái ra.

Thực hành trên máy tính: Giáo viên cho học sinh thực hành kéo thả 4 câu lệnh ở Bảng 1 vào chương trình như Hình 3.



Hình 3: Kéo thả câu lệnh vào khung chương trình

3. Thay đổi thông số cho câu lệnh (5 phút)

Trong các câu lệnh Scratch, những vị trí được tô màu trắng hoặc có dấu tam giác quay xuống, là những vị trí mà chúng ta có thể thay đổi các giá trị bên trong khi viết chương trình. Để thay đổi thông số của một lệnh, ví dụ như câu lệnh **di chuyển 10 bước**, chúng ta nhấp đôi chuột vào ô màu trắng. Dấu nháy chuột sẽ hiện ra và cho phép chúng ta thay đổi giá trị.

Thực hành trên máy tính: Giáo viên cho học sinh thay đổi để nhân vật di chuyển 20 bước, xoay sang phải 30 độ và xoay sang trái 30 độ. Kết quả đạt được như hình sau:



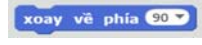
Hình 4: Các câu lệnh sau khi đã thay đổi thông số

Thực hành trên máy tính: Giáo viên cho học sinh nhấp chuột vào câu lệnh mới, để xem tác dụng của các câu lệnh sau khi đã thay đổi thông số. Kết quả đạt được: Nhân vật sẽ di chuyển xa hơn gấp đôi (20 bước so với 10 bước ban đầu) và xoay nhiều hơn gấp đôi (30 độ so với 15 độ ban đầu).

Đối với câu lệnh có dấu tam giác hướng xuống, chúng ta chỉ cần nhấp chuột vào đó, một danh sách sẽ hiện ra cho chúng ta lựa chọn như minh họa ở hình dưới đây:



Hình 5: Thay đổi thông số bằng cách lựa chọn trong danh sách có sẵn

Thực hành trên máy tính: Giáo viên cho học sinh thay đổi thông số của câu lệnh . Nhấp chuột vào câu lệnh này để thấy sự thay đổi của nhân vật.

4. Tạo bản sao cho một câu lệnh (5 phút)

Trong một số trường hợp, thay vì kéo thả câu lệnh ra khung chương trình và thay đổi thông số (nếu có), chúng ta có nhu cầu tạo một bản sao từ câu lệnh có sẵn. Để làm được điều này, chúng ta **nhấp chuột phải** vào câu lệnh đó, và chọn **tạo bản sao**.

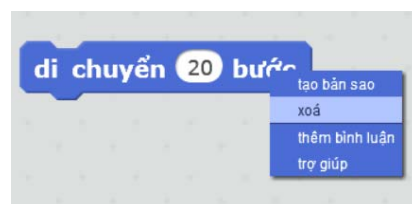


Hình 6: Tạo bản sao cho một câu lệnh: Nhấp chuột phải và chọn tạo bản sao

Sau khi chọn tạo bản sao, chúng ta **di chuyển chuột tới vị trí mới** và **nhấn chuột trái** để đặt câu lệnh vừa được tạo ra.

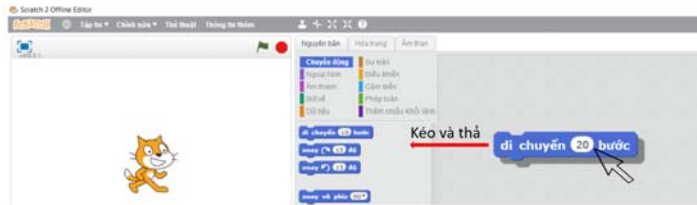
5. Xóa một câu lệnh (5 phút)

Tương tự như việc tạo bản sao cho một câu lệnh, chúng ta có thể **nhấn chuột phải** và chọn **xóa** khi không muốn dùng câu lệnh này nữa.



Hình 7: Xóa một câu lệnh: Nhấp chuột phải và chọn xóa

Một cách khác để xóa một câu lệnh là kéo nó từ khung chương trình, sau đó thả vào danh sách các câu lệnh (Xem minh họa ở Hình 8).



Hình 8: Xóa một câu lệnh bằng cách kéo và thả nó từ chương trình vào danh sách lệnh

6. Ghép nối các câu lệnh (5 phút)

Đây là tác vụ quan trọng nhất khi làm việc với Scratch. Việc viết một chương trình thay vì phải gõ từng dòng lệnh như ngôn ngữ lập trình truyền thống (ví dụ: Pascal), chúng ta chỉ cần ghép nối các câu lệnh lại với nhau. Để làm được việc này, chúng ta chỉ việc **kéo** câu lệnh gần với nhau, một kí hiệu màu trắng xuất hiện, sau đó chúng ta có thể **thả**, để ghép các câu lệnh lại với nhau.



Hình 9: Kéo câu lệnh gần nhau, khi thấy kí hiệu màu trắng, thả ra để ghép chúng lại

Một lưu ý quan trọng, khi ghép nhiều câu lệnh lại với nhau, chương trình sẽ xem đó như một câu lệnh. Do đó, các tác vụ như kéo thả, tạo bản sao hay xóa sẽ có tác dụng trên tất cả các câu lệnh được ghép.

7. Tách các câu lệnh (5 phút)

Ngược lại với việc ghép các câu lệnh, là tách các câu lệnh ra. **Một lưu ý quan trọng là chúng ta chỉ có thể tách các câu lệnh rời ra từ bên dưới.** Để tách câu lệnh, chúng ta thực hiện tác vụ kéo-thả câu lệnh cần tách ra, di chuyển chuột xuống phía dưới, cho đến khi kí hiệu màu trắng mất đi.



Hình 10: Kéo câu lệnh cần tách xuống phía dưới, cho đến khi kí hiệu màu trắng mất đi

8. Bài tập trên lớp (10 phút)

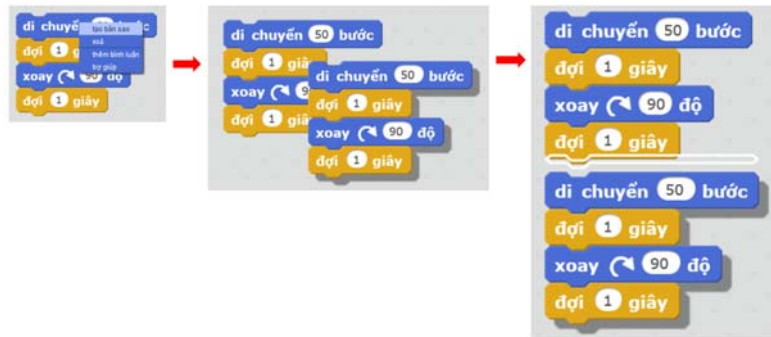
Giáo viên hướng dẫn học sinh cho mèo di chuyển thành hình vuông. **Trong quá trình thực hành, nhấn vào lệnh `xoay về phía 90` để điều chỉnh lại nhân vật nếu cần thiết.**

Bước 1: Cho mèo di chuyển 1 cạnh (50 bước), sau đó xoay 90 độ về phía bên phải. Để có thể thấy hiệu ứng di chuyển rồi xoay, chúng ta sẽ chèn câu lệnh **đợi 1s** vào giữa 2 câu lệnh. Câu lệnh này có màu vàng và nằm trong nhóm lệnh **Điều khiển**



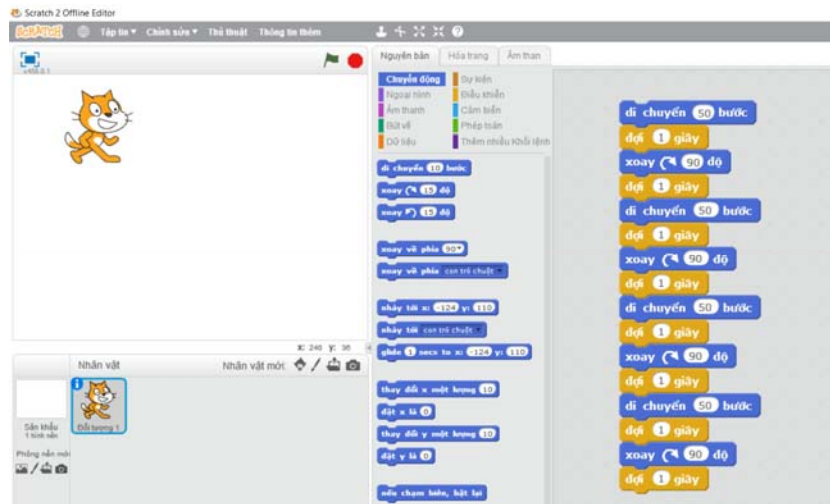
Hình 11: Di chuyển theo 1 cạnh của hình vuông

Bước 2: Giáo viên hướng dẫn cho học sinh cách tạo bản sao cho tất cả 4 lệnh trên: **Nhấp chuột phải vào câu lệnh đầu tiên của nhóm 4 câu lệnh** và chọn tạo bản sao, di chuyển bản sao gồm 4 câu lệnh đến vị trí thích hợp (kí hiệu màu trắng xuất hiện) sau đó nhấn chuột trái để ghép chúng lại với nhau:



Hình 12: Tạo bản sao cho nhóm 4 câu lệnh và ghép chúng lại với nhau

Bước 3: Làm tương tự như bước 2, tạo bản sao và ghép 8 câu lệnh mới vào câu lệnh cũ. Như vậy chúng ta đã hoàn thành chương trình mèo di chuyển theo hình vuông. Nhấp chuột vào bất kì lệnh nào trong chương trình, nhân vật của chúng ta sẽ bắt đầu di chuyển.



Hình 13: Chương trình mèo di chuyển theo hình vuông hoàn chỉnh

9. Bài tập về nhà (Tùy thuộc vào điều kiện của NT)

- Học sinh hãy hiện thực chương trình mèo di chuyển theo hình tam giác.

Đáp án: Mỗi lần di chuyển xoay góc 120 độ, lặp lại 3 lần



Hình 14: Chương trình mèo đi chuyển theo hình tam giác

2. Học sinh hãy hiện thực chương trình mèo đi chuyển theo hình lục giác.
Đáp án: Mỗi lần đi chuyển xoay góc 60 độ, lặp lại 6 lần

TIẾT 3:

VẼ HÌNH TRÊN SCRATCH

MỤC TIÊU

- Học sinh nắm được cách vẽ hình trên Scratch.
- Học sinh có khả năng hoàn thiện chương trình Scratch.
- Học sinh có khả năng lưu và mở chương trình.

PHÂN BỐ THỜI LƯỢNG

1. Sửa bài tập (5 phút)
2. Hoàn thiện chương trình Scratch (5 phút)
3. Vẽ hình trên Scratch (15 phút)
4. Lưu và Mở chương trình (10 phút)
5. Bài tập trên lớp (10 phút)
6. Bài tập về nhà

1. Sửa bài tập

(5 phút)

Giáo viên sửa bài tập về nhà ở Bài 2: Cho mèo đi chuyển theo hình tam giác. Đối với các học sinh chưa hiện thực được, giáo viên cho xem chương trình mẫu sau đây để các em tự hiện thực lại:






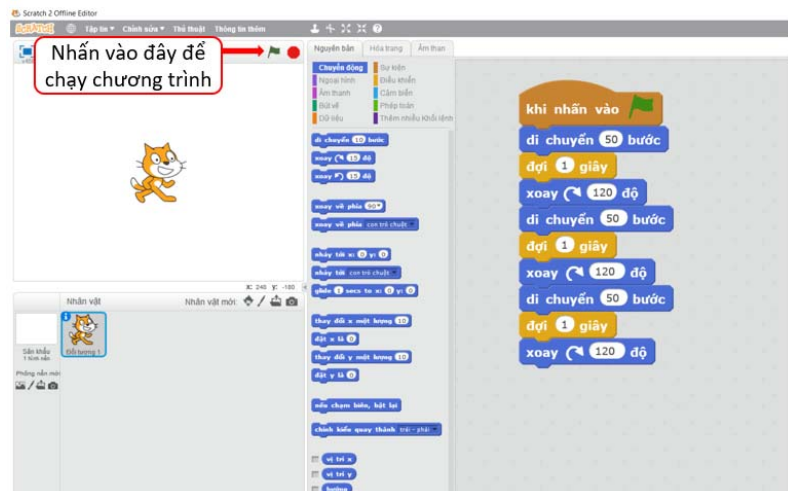
Hình 1: Chương trình mèo đi chuyển theo hình tam giác

2. Hoàn thiện chương trình Scratch

(5 phút)

Hiện tại, để thực thi chương trình Scratch, chúng ta phải nhấp chuột vào các câu lệnh được ghép nối bên cửa sổ chương trình. Scratch hỗ trợ một số sự kiện để chương trình có thể thực thi mà không cần nhấp chuột vào chương trình. Các câu lệnh này thuộc nhóm **Sự kiện**.

Đầu tiên là câu lệnh sự kiện , khi nhấp vào biểu tượng , chương trình sẽ thực thi. Giáo viên hướng dẫn học sinh chèn thêm câu lệnh này vào đầu chương trình mèo đi chuyển theo hình tam giác. Bây giờ khi nhấp vào biểu tượng , chú mèo Scratch sẽ di chuyển.


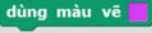
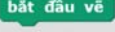


Hình 2: Thêm sự kiện là cờ để thực thi chương trình

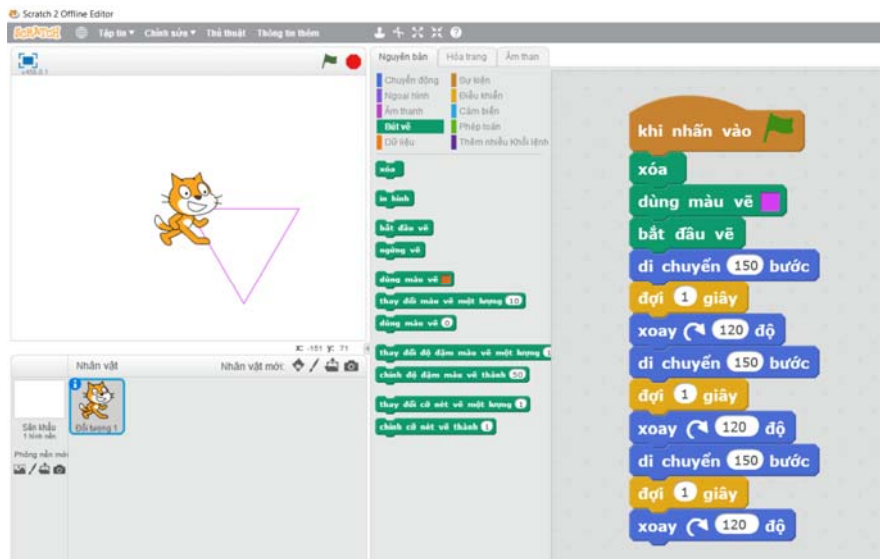
3. Vẽ hình trên Scratch

(15 phút)



Nguyên lý vẽ hình trên Scratch rất đơn giản: Nét vẽ sẽ đi theo đường di chuyển của nhân vật. Để làm được tác vụ này, chúng ta cần phải kích hoạt chức năng vẽ hình trên Scratch. Ba câu lệnh sẽ được sử dụng để thực hiện chức năng này. Chúng đều nằm trong nhóm lệnh Bút vẽ, màu xanh lá cây như sau:

Câu lệnh	Chức năng
	Xóa những nét vẽ trước đó, để bắt đầu vẽ hình mới
	Thiết lập màu vẽ
	Bắt đầu vẽ, sau câu lệnh này, nhân vật di chuyển đến đâu, sẽ vẽ hình ra đến đấy

Chương trình hoàn thiện và kết quả được mô tả như hình dưới đây:



Hình 3: Một hình tam giác được vẽ ra khi mèo di chuyển

Để thay đổi màu vẽ, chúng ta nhấp chuột trái vào màu đang sử dụng của câu lệnh , con trỏ chuột sẽ thay đổi thành hình . Sau đó, di chuyển chuột đến một màu mong muốn nào đó có sẵn trên màn hình, nhấp chuột để thay đổi màu cho bút vẽ. Hình 4 là minh họa về trình tự để thay đổi màu vẽ.

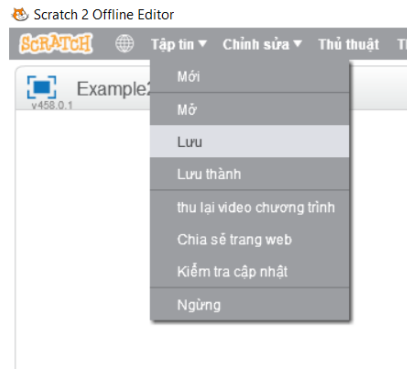


Hình 4: Trình tự thay đổi màu vẽ: Nhấp chuột trái vào màu hiện tại, di chuyển chuột tới màu mới, nhấp chuột trái thêm lần nữa để chọn màu

4. Lưu và Mở chương trình

(10 phút)

Lưu chương trình là bước cần thiết để chúng ta có thể tiếp tục phát triển cho lần sau. Để lưu chương trình, chúng ta chọn Tập tin, chọn Lưu, chọn đường dẫn phù hợp, đặt tên cho file và nhấn OK.



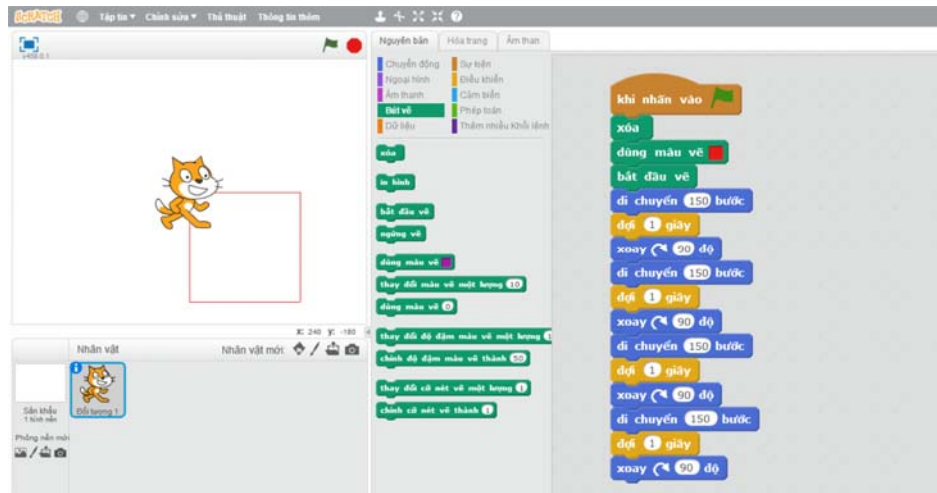
Hình 5: Lưu và Mở lại chương trình

Tương tự, để Mở lại file vừa lưu, chúng ta cũng vào Tập tin, chọn Mở, và chọn đường dẫn đến file đã lưu trước đó.

5. Bài tập trên lớp

(10 phút)

Học sinh hiện thực chương trình vẽ hình vuông bằng cách sử dụng lại chương trình mèo di chuyển theo hình vuông đã học ở bài trước. Lưu lại chương trình trong máy tính với tên **vehinhvuong.sb2**



Hình 6: Chương trình mèo vẽ hình vuông hoàn chỉnh

6. Bài tập về nhà

(Tùy vào điều kiện của NT)

Vẽ hình lục giác 6 cạnh bằng ngôn ngữ Scratch. Lưu lại chương trình và trình bày ở lớp ở tiết học tiếp theo.

TIẾT 4:

VÒNG LẶP TRONG SCRATCH

• MỤC TIÊU

- Học sinh hiểu được nguyên lý của vòng lặp.
- Học sinh nắm được cách sử dụng vòng lặp trong Scratch.
- Học sinh có khả năng hiện thực một chương trình cho sẵn.

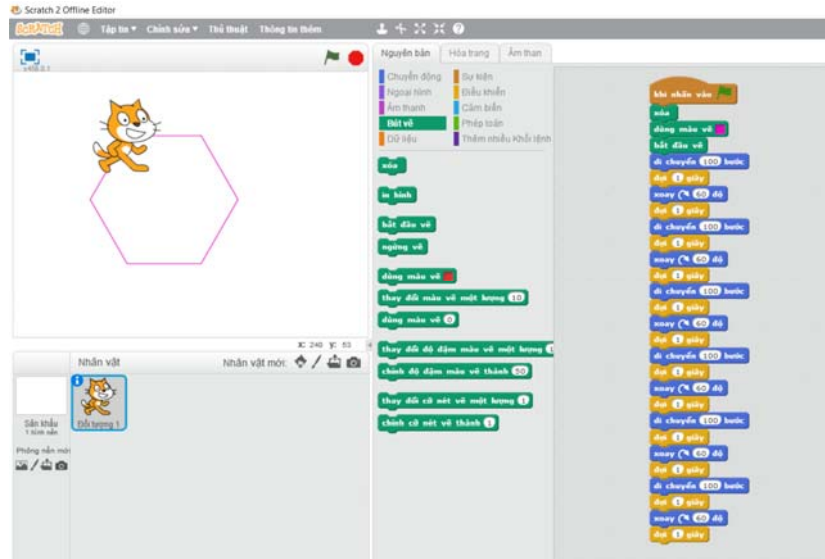
• PHÂN BỐ THỜI LƯỢNG

1. Sửa bài tập (15 phút)
2. Giới thiệu vòng lặp (15 phút)
3. Bài tập trên lớp (15 phút)
4. Bài tập về nhà
5. Bài tập nâng cao

1. Sửa bài tập (5 phút)

Giáo viên sửa bài tập về nhà: Vẽ hình lục giác đều 6 cạnh.

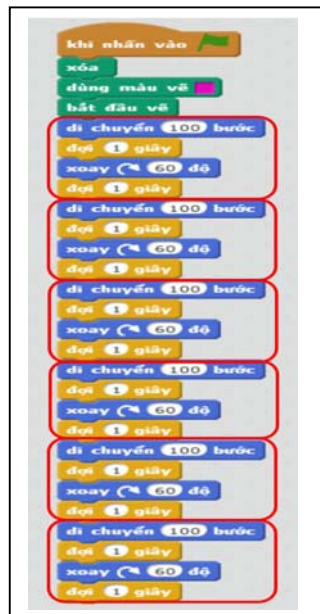
Đáp án: Mỗi lần di chuyển xoay 1 góc 60 độ, lặp lại 6 lần. Chương trình gợi ý như sau



Hình 1: Chương trình vẽ hình lục giác đều 6 cạnh

2. Giới thiệu vòng lặp (15 phút)

Như chương trình ở trên, để vẽ hình lục giác đều, chúng ta cần phải lặp đi lặp lại 1 nhóm 4 câu lệnh như Hình 2. Các hiện thực như hiện tại tương đối hiệu, tuy nhiên lại làm chương trình khá dài và rất khó để thay đổi cũng như mở rộng.



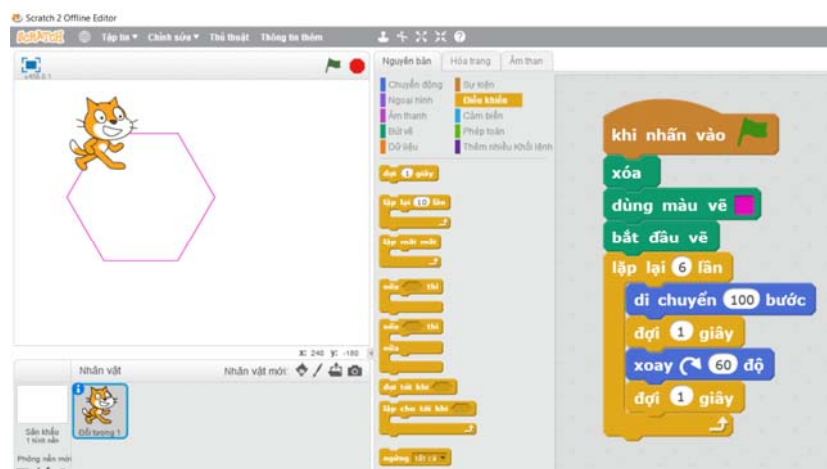
Hình 2: Các câu lệnh được lặp lại 6 lần khi vẽ hình lục giác

Để khắc phục vấn đề này, Scratch hỗ trợ câu lệnh lặp lại. Câu lệnh này nằm trong nhóm lệnh Điều khiển, có màu vàng, như mô tả ở Hình 3.



Hình 3: Câu lệnh lặp lại trong Scratch

Như chúng ta có thể thấy, mặc định câu lệnh này sẽ lặp lại 10 lần những gì được ghép bên trong nó. Do đó, chúng ta chỉ cần hiện thực một lần lặp và thay đổi giá trị **10** thành số lần lặp mà chúng ta mong muốn. Ví dụ, chương trình vẽ hình lục giác đều được hiện thực lại với 6 lần lặp như sau:



Hình 4: Vẽ hình lục giác bằng cách sử dụng vòng lặp

Chúng ta có thể thấy, chương trình được rút gọn đi rất nhiều. Và như đã trình bày ở trên, chương trình sử dụng vòng lặp có thể được thay đổi và mở rộng rất nhanh chóng.

3. Bài tập trên lớp

(15 phút)

Bài 1: Hiện thực vẽ hình vuông sử dụng vòng lặp

Đáp án: Sử dụng 4 lần lặp, mỗi lần xoay góc 90 độ. Chúng ta chỉ cần thay đổi số lần lặp và góc xoay ở chương trình vẽ lục giác, là có được hình vuông.



Hình 5: Vẽ hình vuông bằng vòng lặp

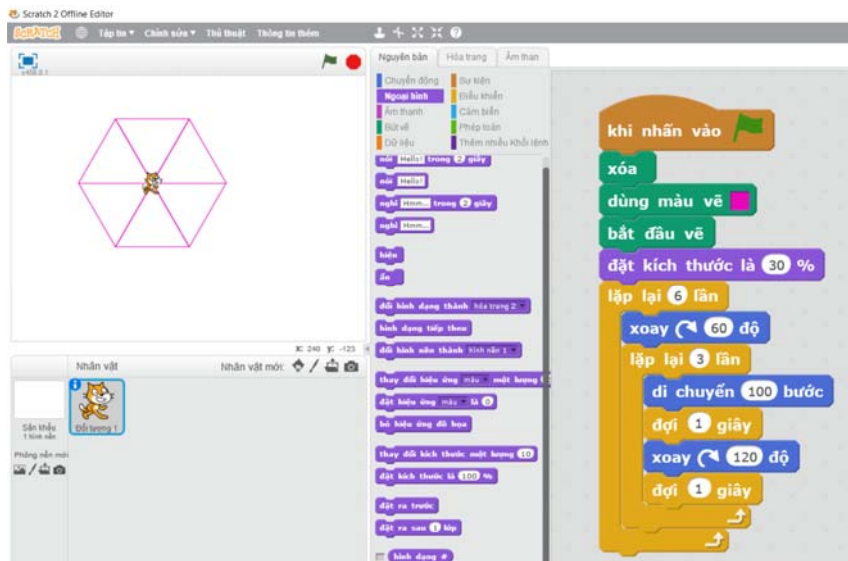
Bài 2: Hiện thực vẽ hình tam giác sử dụng vòng lặp

Đáp án: Sử dụng 3 lần lặp, mỗi lần xoay góc 120 độ. Chương trình gợi ý như sau:



Hình 6: Chương trình vẽ hình tam giác bằng vòng lặp

Bài 3: Giáo viên cho học sinh xem một chương trình mẫu và yêu cầu hiệu thực lại, ví dụ như sau:



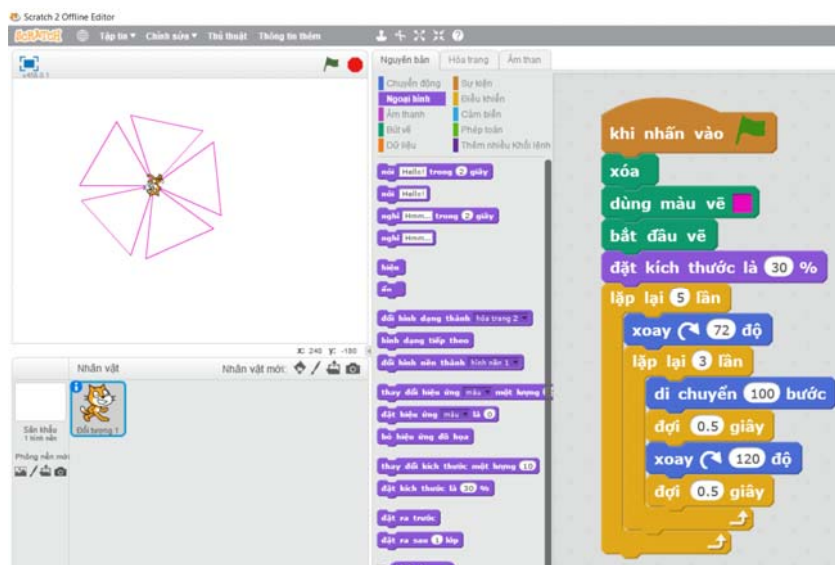
Hình 7: Chương trình mẫu vẽ hình trên Scratch

Thông qua bài tập này, giáo viên lưu ý lại với học sinh về màu sắc của mỗi câu lệnh để tìm kiếm theo nhóm. Trong chương trình này, có câu lệnh mới, **đặt kích thước là 30 %**, câu lệnh này có màu tím và nằm trong nhóm Ngoại hình, có tác dụng thu nhỏ hoặc phóng to nhân vật của chúng ta.

4. Bài tập về nhà

(Tùy điều kiện NT)

Giáo viên yêu cầu học sinh tìm kiếm những chương trình vẽ hình trên mạng để tham khảo và tự hiện thực lại bằng cách quan sát màu của các câu lệnh. Một ví dụ khác như sau:

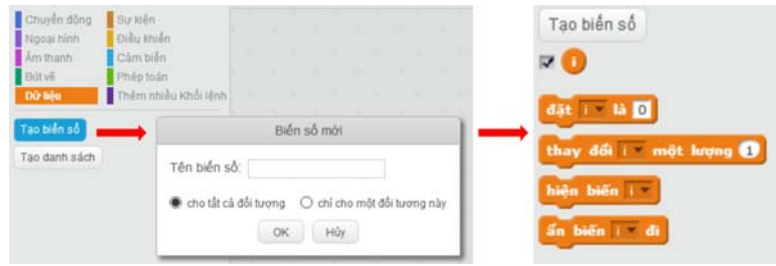


Hình 8: Vẽ hình trên Scratch

5. Bài tập về nhà nâng cao

(Tùy điều kiện NT)

Để có thể vẽ được nhiều hoa văn đẹp, chúng ta cần tạo thêm 1 biến số có giá trị tăng dần sau mỗi vòng lặp. Để tạo được biến số, chúng ta vào nhóm câu lệnh **Dữ liệu**, nhấn vào nút **Tạo biến số** như Hình 9.



Hình 9: Trình tự tạo biến số trên Scratch

Một cửa sổ hiện lên để chúng ta nhập tên bên số. Sau khi tạo biến số thành công, một số mảnh ghép mới sẽ được tự động sinh ra. Như ví dụ ở Hình 9, 4 mảnh ghép cho biến i được sinh ra. Từ đây, chúng ta có thể hiện thực được rất nhiều chương trình vẽ hoa văn đẹp như sau:



Hình 10: Chương trình vẽ hoa văn dùng vòng lặp

Lưu ý: Trong chương trình trên, để hiện thực câu lệnh **đi chuyển i bước**, chúng ta phải ghép câu lệnh **đi chuyển 10 bước** và biến i (Xem minh họa ở Hình 11).



Hình 11: Ghép câu lệnh đi chuyển 10 bước và biến i

TIẾT 5:

LẬP TRÌNH HOẠT HÌNH TRONG SCRATCH

MỤC TIÊU

- Học sinh sử dụng được các lệnh tạo hiệu ứng di chuyển.
- Học sinh nắm được cách tạo một hoạt cảnh hoạt hình đơn giản.
- Học sinh có khả năng hiện thực một chương trình cho sẵn.

PHÂN BỐ THỜI LƯỢNG

1. Sửa bài tập (10 phút)
2. Thay đổi cảnh nền trong hoạt cảnh hoạt hình (10 phút)
3. Giới thiệu lệnh tạo hiệu ứng di chuyển. (15 phút)
4. Bài tập trên lớp (10 phút)
5. Bài tập về nhà

1. Sửa bài tập

(10 phút)

Giáo viên cho học sinh trình bày bài tập về nhà của học sinh (nếu có) và giải thích nguyên lý vẽ hình hoa văn trong đoạn chương trình mẫu ở bài trước:

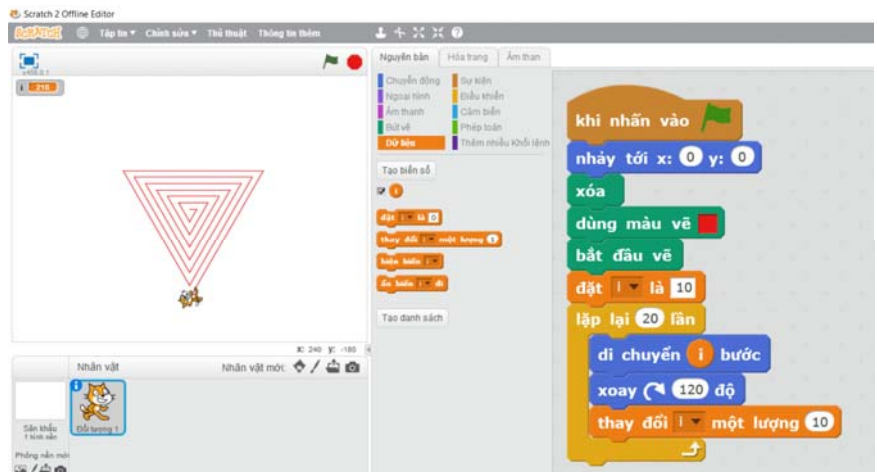


Hình 1: Chương trình vẽ hoa văn bằng Scratch

Để giải thích nguyên lý của bài này, giáo viên cho thay đổi các thông số sau:

1. Đặt giá trị ban đầu của i là 10
2. Lặp 20 lần
3. Mỗi lần lặp tăng giá trị của i lên 10 đơn vị
4. Chỉnh góc xoay lại thành 120 độ

Chương trình mới với kết quả như sau:



Hình 2: Chương trình vẽ hoa văn đơn giản

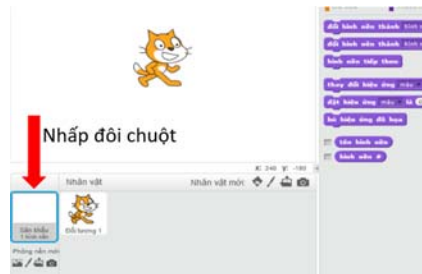
Như vậy, chúng ta đang vẽ từng cạnh của hình tam giác, tuy nhiên chiều dài của mỗi cạnh tăng dần lên. Khi đặt giá trị ban đầu cho i là 1 đơn vị, những hình tam giác ban đầu sẽ rất nhỏ. Mỗi lần lặp tăng i lên 1 đơn vị,

các hình tam giác sẽ gần nhau hơn. Cuối cùng, thay vì xoay góc 120 độ để vẽ hình tam giác, chúng ta cho 1 giá trị khác đi, để tạo cảm giác là hình bị xoay và tạo ra hoa văn. Trong trường hợp vẽ hoa văn ở Hình 1, chúng ta cho xoay 122 độ sau mỗi lần vẽ.

2. Thay đổi cảnh nền trong hoạt cảnh hoạt hình (10 phút)

Để có thể tạo những hoạt cảnh hoạt hình đẹp, chúng ta có thể chọn một hình nền có sẵn do Scratch hỗ trợ. Các bước để thay đổi hình nền như sau:

Bước 1: Nhấp đôi vào biểu tượng Sân khấu (sẽ được tô viền màu xanh)



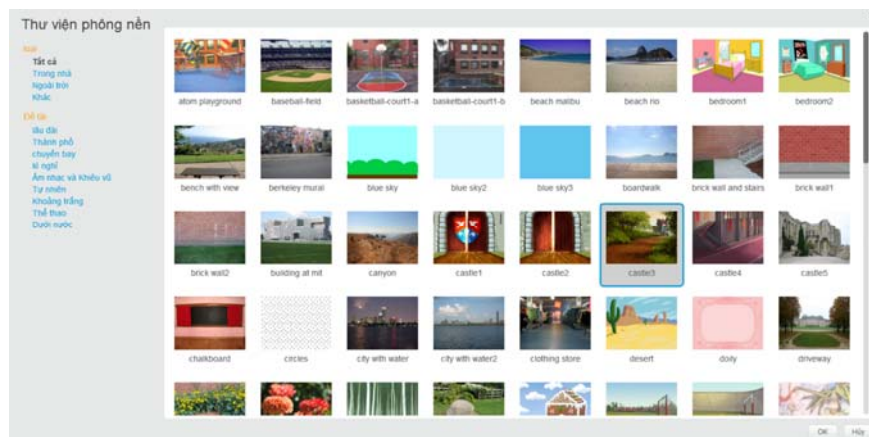
Hình 3: Nhấp đôi vào biểu tượng sân khấu ở góc dưới bên trái

Bước 2: Nhấp chuột vào biểu tượng chọn phong nền từ thư viện.



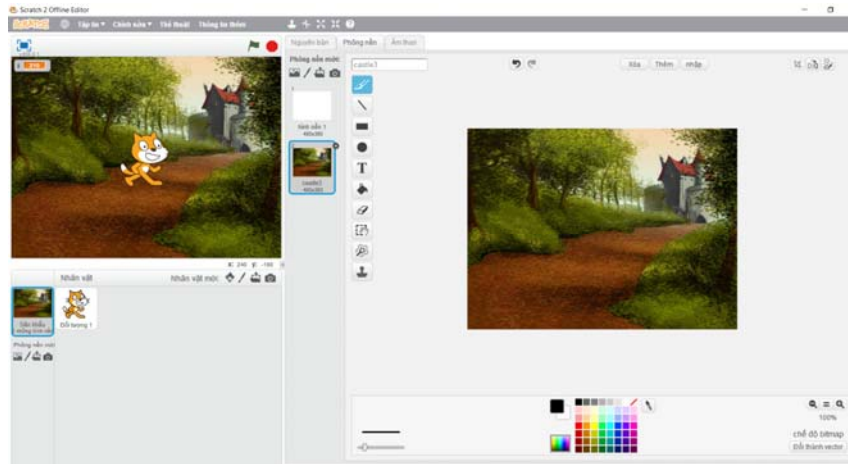
Hình 4: Nhấp chuột vào biểu tượng Chọn phong nền từ thư viện

Bước 3: Một cửa sổ với nhiều phong nền có sẵn sẽ hiện ra. Để thống nhất, giáo viên yêu cầu học sinh chọn hình nền có tên castle3



Hình 5: Chọn hình nền castle3

Bước 4: Nhấn nút OK, và hình nền mới sẽ được đưa vào chương trình như sau:



Hình 6: Hình nền được cập nhật

3. Giới thiệu lệnh tạo hiệu ứng di chuyển (15 phút)

Trong bài học này, chúng ta sẽ sử dụng các lệnh sau đây để tạo một hoạt cảnh hoạt hình trên Scratch.

Câu lệnh	Chức năng
<code>nhảy tới x: 0 y: 0</code>	Mang nhân vật tới vị trí có tọa độ [x,y] trong sân khấu. Tọa độ [0,0] là vị trí trung tâm sân khấu.
<code>glide 1 secs to x: 50 y: 50</code>	Trượt nhân vật từ vị trí hiện tại tới vị trí [x,y], thời gian trượt mặc định là 1s và có thể thay đổi được.

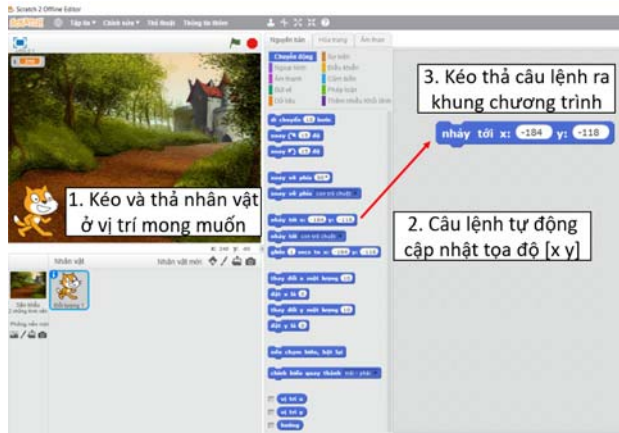
Bảng 1: Các câu lệnh tạo hiệu ứng chuyển động

Để bắt đầu lập trình, chúng ta cần nhấp đôi để chọn nhân vật là chú mèo Scratch:



Hình 7: Chọn lại nhân vật, chọn Nguyên bản

Để sử dụng hiệu quả 2 câu lệnh trong bài này, chúng ta làm 3 bước như minh họa ở Hình 8:



Hình 8: Ba bước để thực hiện 2 câu lệnh di chuyển

Khi thực hiện theo trình tự trên, chúng ta không cần thay đổi các giá trị [x,y]. Thay vào đó, khi kéo thả nhân vật tới vị trí mong muốn, chương trình sẽ tự động cập nhật giá trị cho chúng. Ở Hình 8, chúng ta kéo thả nhân vật vào góc dưới bên trái của hoạt cảnh, và kéo câu lệnh `nhảy tới x: 0 y: 0` (lúc này x và y đã được cập nhật) ra khung chương trình. Như vậy, khi chương trình chạy, nhân vật của chúng ta sẽ luôn bắt đầu từ vị trí ở góc cảnh nền.

Tương tự, chúng ta kéo và thả nhân vật mèo tới vị trí mới, sau đó kéo câu lệnh `glide 1 secs to x: 50 y: 50` ra màn hình (các giá trị [x,y] sẽ được cập nhật tương ứng). Chúng ta cũng thêm sự kiện chạy chương trình khi nhấn vào lá cờ. Chương trình sẽ như sau:



Hình 9: Chương trình mèo di chuyển khi nhấn vào lá cờ

4. Bài tập trên lớp (10 phút)

Giáo viên cho học sinh hiện thực chương trình ở Hình 9. Sau đó yêu cầu các em làm thêm những đoạn di chuyển khác và cuối cùng dừng lại ở lâu đài. Chúng ta có thể cho nhân vật thông báo lên câu “Tới nơi rồi”. Học sinh lưu lại chương trình trong máy tính. Chương trình gợi ý như sau:



Hình 10: Chương trình mèo di chuyển tới lâu đài

5. Bài tập về nhà

(Tùy vào điều kiện NT)

Học sinh chọn một hoạt cảnh khác, và cho nhân vật di chuyển trong hoạt cảnh theo kịch bản của riêng mình.

TIẾT 6:

LẬP TRÌNH HOẠT HÌNH TRONG SCRATCH (TIẾP THEO)

• MỤC TIÊU

- Học sinh sử dụng được các lệnh tạo hiệu ứng đi chuyên.
- Học sinh nắm được cách tạo nhiều hoạt cảnh hoạt hình.
- Học sinh có khả năng hiện thực một chương trình cho sẵn.

• PHÂN BỐ THỜI LƯỢNG

1. Sửa bài tập (10 phút)
2. Thêm đổi cảnh nền (10 phút)
3. Đổi cảnh nền trong hoạt cảnh (10 phút)
4. Bài tập trên lớp (15 phút)
5. Bài tập về nhà

1. Sửa bài tập (10 phút)

Giáo viên cho học sinh trình bày các hoạt cảnh hoạt hình các em tự làm ở nhà (nếu có).

Giáo viên cho học sinh mở lại chương trình hoạt cảnh đã hiện thực ở bài trước.

2. Thêm đổi cảnh nền (10 phút)

Ở bài trước, chúng ta đã lập trình cho nhân vật di chuyển trong một cảnh nền. Trước tiên, chúng ta hãy mở lại chương trình đã hiện thực ở bài trước.



Hình 1: Chương trình mèo di chuyển ở Bài 5

Trong bài này, chúng ta sẽ cho nhân vật di chuyển trong 2 cảnh nền khác nhau. Để làm được việc này, trước hết chúng ta phải thêm một cảnh nền khác. Các bước thêm cảnh nền được trình bày bên dưới, và hoàn toàn giống với bài trước. Tuy nhiên ở bài này, chúng ta sẽ thêm cảnh nền thứ 2, có tên là **castle4**.

Bước 1: Nhấp đôi vào biểu tượng Sân khấu (sẽ được tô viền màu xanh)



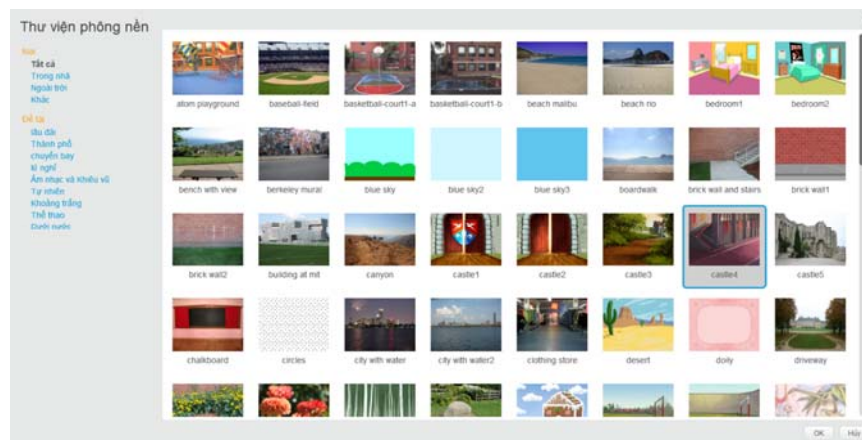
Hình 2: Nhấp đôi vào biểu tượng sân khấu ở góc dưới bên trái

Bước 2: Nhấp chuột vào biểu tượng chọn phong nền từ thư viện.



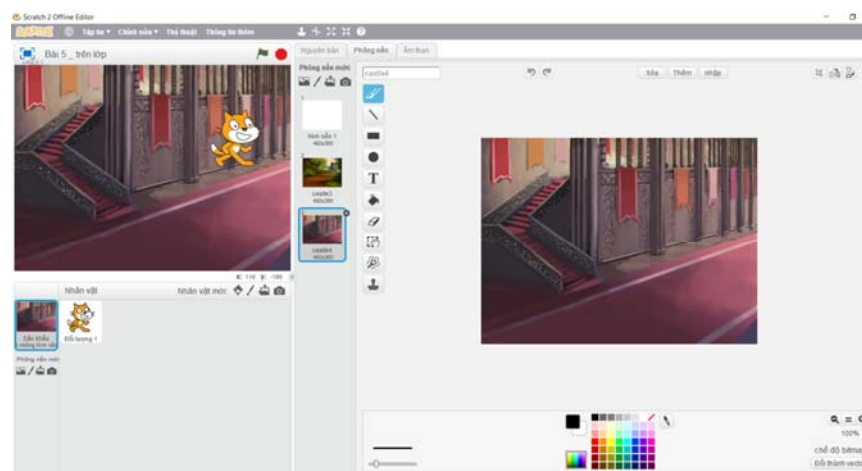
Hình 3: Nhấp chuột vào biểu tượng Chọn phòng nền từ thư viện

Bước 3: Một cửa sổ với nhiều phòng nền có sẵn sẽ hiện ra. Để thống nhất, giáo viên yêu cầu học sinh chọn hình nền có tên **castle3**



Hình 4: Chọn hình nền castle4

Bước 4: Nhấn nút OK, và hình nền mới sẽ được đưa vào chương trình như sau:



Hình 5: Hình nền mới được thêm vào, chúng ta có 2 hình nền

3. Đổi cảnh nền trong hoạt cảnh (10 phút)

Trong bài học này, chúng ta sẽ sử dụng câu lệnh sau đây để thay đổi ảnh nền trong hoạt cảnh hoạt hình trên Scratch. Câu lệnh này nằm trong nhóm các câu lệnh về Ngoại hình, có màu tím.

Câu lệnh	Chức năng
đổi hình nền thành castle4	Đổi ảnh nền trong các ảnh nền có sẵn. Lựa chọn ảnh nền bằng dấu tam giác hướng xuống

Bảng 1: Câu lệnh thay đổi ảnh nền

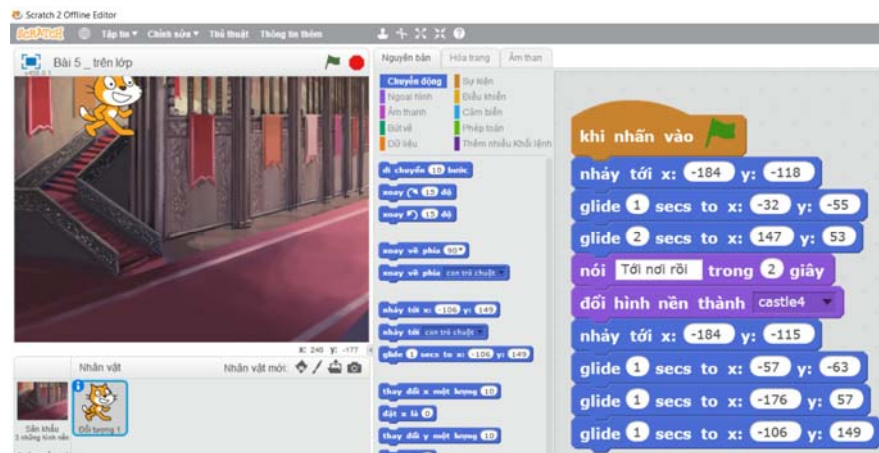
Để bắt đầu lập trình, chúng ta cần nhấp đôi để chọn nhân vật là chú mèo Scratch:



Hình 6: Chọn lại nhân vật, chọn Nguyên bản

Tiếp theo, chúng ta ghép câu lệnh `đổi hình nền thành castle4`. Chúng ta thấy rằng, khi đổi cảnh nền, vị trí của nhân vật chính là vị trí cuối cùng ở cảnh nền trước đó (castle3). Vì vậy, chúng ta sẽ sử dụng lại câu lệnh `nhảy tới x: 0 y: 0`, để thiết lập lại vị trí ban đầu cho nhân vật của chúng ta ở cảnh nền mới. Sau đó tiếp tục sử dụng các câu lệnh trượt `glide 1 secs to x: 50 y: 50` để tạo hiệu ứng di chuyển cho nhân vật.

Giáo viên nhắc lại cho học sinh, để sử dụng 2 câu lệnh này, chúng ta phải di chuyển nhân vật đến vị trí mong muốn trước, để các giá trị [x,y] được cập nhật, sau đó mới kéo câu lệnh ra màn hình. Đến bước này, chương trình của chúng ta như sau:



Hình 7: Chương trình mèo di chuyển trong hoạt cảnh thứ 2

Tuy nhiên, để đảm bảo mỗi khi chạy chương trình, hoạt cảnh đầu tiên (castle3) sẽ được hiện ra, chúng ta cần chỉnh lại hình nền khi vừa chạy chương trình. Chương trình hoàn chỉnh như sau:



Hình 8: Chương trình hoàn chỉnh - Đổi cảnh nền castle3 khi vừa chạy chương trình

4. Bài tập trên lớp

(15 phút)

Giáo viên cho học sinh hiện thực chương trình ở Hình 8. Sau đó yêu cầu các em làm thêm những hoạt động khác như phát ra tiếng mèo kêu hoặc cho nhân vật của chúng ta xoay mặt qua hướng trái phải mỗi khi leo lên cầu thang. Chương trình gợi ý như sau:



Hình 9: Một ví dụ về hoạt cảnh hoạt hình

5. Bài tập về nhà

(Tùy điều kiện NT)

Học sinh chọn thêm các hoạt cảnh khác và xây dựng các hiệu ứng di chuyển như hướng dẫn ở bài này.

TIẾT 7:

LẬP TRÌNH NHẬP/XUẤT TRONG SCRATCH

• MỤC TIÊU

- Học sinh sử dụng được lệnh nhận dữ liệu từ người dùng.
- Học sinh sử dụng được lệnh xuất dữ liệu thông báo ra màn hình.
- Học sinh làm được chương trình nhập/xuất đơn giản.

• PHÂN BỐ THỜI LƯỢNG

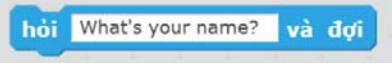

1. Sửa bài tập (10 phút)
2. Nhập dữ liệu (15 phút)
3. Xuất dữ liệu (15 phút)
4. Bài tập trên lớp (5 phút)

1. Sửa bài tập (10 phút)

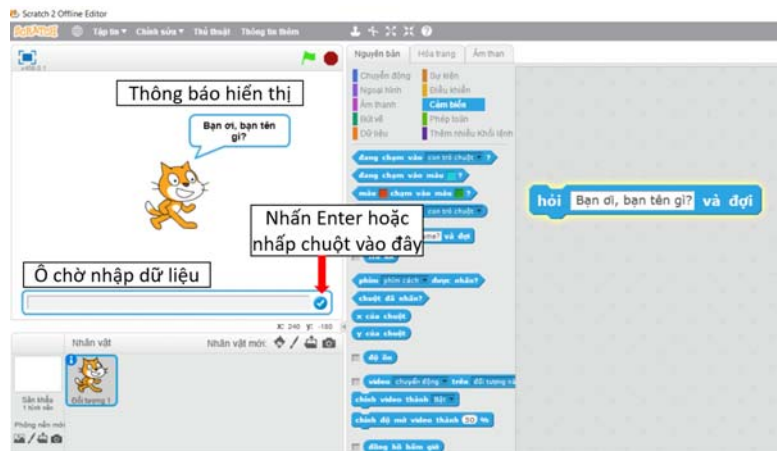
Chọn một bài tập tiêu biểu của học sinh và trình chiếu cho cả lớp cùng xem.

2. Nhập dữ liệu (15 phút)


Trong một số ứng dụng, thông thường chương trình sẽ yêu cầu người dùng nhập thông tin. Tùy vào thông tin của người dùng nhập vào, mà chương trình sẽ xử lý tương ứng. Ví dụ, một chương trình có thể yêu cầu người dùng nhập hai số a và b, sau đó tính tổng của hai số này và xuất kết quả ra màn hình. Trong phần này, chúng ta sẽ tìm hiểu câu lệnh để chương trình yêu cầu người dùng nhập dữ liệu vào. Câu lệnh này có màu xanh dương, và nằm trong nhóm **Cảm biến**.

Câu lệnh	Tác dụng
	Nhân vật sẽ nói ra thông báo What's your name? ra màn hình, đồng thời một khung nhập liệu sẽ hiển thị ra bên dưới nhân vật, chờ người dùng nhập dữ liệu vào. Câu lệnh này sẽ kết thúc khi người dùng nhấn phím Enter trên bàn phím, hoặc nhấn vào dấu  ở bên phải khung nhập liệu

Trong câu lệnh trên, chúng ta hoàn toàn có thể thay đổi nội dung thông báo. Trong ví dụ dưới đây, chúng ta đổi thành câu “Bạn ơi, bạn tên gì?”



Hình 1: Thực thi câu lệnh nhập dữ liệu từ người dùng

Khi người dùng nhấn phím Enter hoặc nhấp chuột vào biểu tượng , tất cả những gì nhập vào sẽ được lưu trong biến tạm **trả lời**. Biến này được hỗ trợ sẵn bởi Scratch và cũng có màu xanh dương, thuộc nhóm các câu lệnh

Cảm biến. Sở dĩ **trả lời** được gọi là biến tạm, vì với lần nhập liệu tiếp theo, tất cả các giá trị trước đó sẽ mất. Để lưu giá trị của biến nhập vào, chúng ta cần phải tạo thêm biến mới.

Để tạo thêm 1 biến số, chúng ta vào nhóm câu lệnh **Dữ liệu**, nhấn vào nút **Tạo biến số** như Hình 2.



Hình 2: Trình tự tạo biến số trên Scratch

Một cửa sổ hiện lên để chúng ta nhập tên bên số, giả sử chúng ta tạo một biến số là **Tên** (biến số trên Scratch có thể đặt có dấu). Sau khi tạo biến số thành công, một số mảnh ghép mới sẽ được tự động sinh ra.

Để lưu giá trị từ người dùng nhập vào (lưu trong biến **trả lời**), chúng ta cần thực hiện như Hình 3: Bằng cách ghép 2 câu lệnh **đặt Tên là 0** và **trả lời**. Chúng ta sẽ kéo thả biến **trả lời** vào ô vuông của câu lệnh **đặt Tên là 0**



Hình 3: Lưu giá trị nhập vào (biến trả lời) vào biến Tên (được tạo ra ở Hình 2)


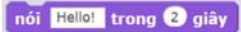
Thực hành trên máy tính: Giáo viên cho học sinh hiện thực một đoạn chương trình ngắn để nhận dữ liệu từ người dùng như Hình 4. Chương trình nhận dữ liệu và lưu vào biến **Tên**



Hình 4: Chương trình nhập dữ liệu và lưu vào biến Tên

3. Xuất dữ liệu (15 phút)

Câu lệnh xuất dữ liệu thông thường sẽ được sử dụng để thông báo kết quả của chương trình ra cho người dùng. Ví dụ, sau khi người dùng nhập tên vào và nhấn Enter như ở bài thực hành trước, chúng ta cho nhân vật nói lên câu “Xin chào bạn ...”, với ... chính là dữ liệu người dùng nhập vào. Câu lệnh mà chúng ta sử dụng thuộc nhóm Ngoại hình:

Câu lệnh	Tác dụng
	Tạo một nội dung cần hiển thị ra màn hình, nội dung này được nối bởi phần tử thứ nhất, là hello , và phần tử thứ 2, là word . Chúng ta hoàn toàn có thể thay đổi phần tử thứ nhất thành câu thông báo theo ý định, chẳng hạn như Xin chào bạn. Đối với phần tử thứ 2, chúng ta có thể thay bằng biến Tên ở bài trên.
	Thông báo ra màn hình nội dung Hello! trong 2 giây. Tuy nhiên, chúng ta sẽ dùng câu lệnh tạo nội dung ở trên để thay thế cho thông báo mặc định này.

4. Bài tập trên lớp

(5 phút)

Thực hành trên máy tính: Giáo viên cho học sinh hiện thực đoạn chương trình sau đây:



Hình 5: Chương trình nhập xuất đơn giản trên Scratch

TIẾT 8:

CÂU LỆNH ĐIỀU KIỆN TRONG SCRATCH

• MỤC TIÊU

- Học sinh hiểu được cấu trúc điều kiện trong ngôn ngữ lập trình
- Học sinh sử dụng được lệnh điều kiện trong Scratch.
- Học sinh làm được chương trình sử dụng lệnh điều kiện đơn giản.

• PHÂN BỐ THỜI LƯỢNG

1. Sửa bài tập (10 phút)
2. Câu lệnh điều kiện (10 phút)
3. Xây dựng điều kiện (10 phút)
4. Bài tập trên lớp – Giáo viên cho học sinh làm bài tập nhóm tự chọn và lưu kết quả. (15 phút)

1. Sửa bài tập

(10 phút)

Đối với bài tập tính hiệu của 2 số A và B trong bài trước, chúng ta chỉ đơn giản là thay phép toán cộng bằng phép toán trừ. Chương trình gọi ý như sau:



Hình 1: Chương trình tính hiệu 2 số A và B

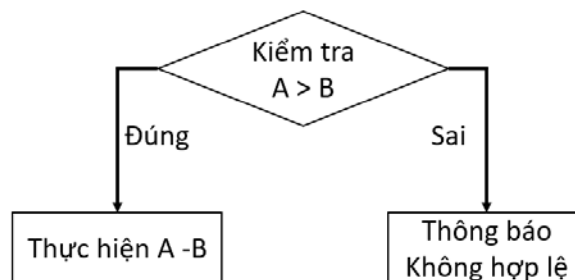
Tuy nhiên, chương trình này cần được cải thiện hơn để hợp lý hơn. Trong trường hợp A nhỏ hơn B, chúng ta sẽ hiện lên thông báo là không thực hiện được phép trừ. Chúng ta chỉ hiện thị kết quả A-B khi A lớn hơn B. Để làm được chức năng này, chúng ta sẽ sử dụng **câu lệnh điều kiện** trong Scratch.

2. Câu lệnh điều kiện

(10 phút)

Câu lệnh điều kiện là câu lệnh được hỗ trợ bởi tất cả các ngôn ngữ lập trình thông dụng. Nguyên lý của câu lệnh này như sau:

1. Kiểm tra một điều kiện nào đó
2. Điều kiện đúng: Thực hiện lệnh 1
3. Điều kiện sai: Thực hiện lệnh 2

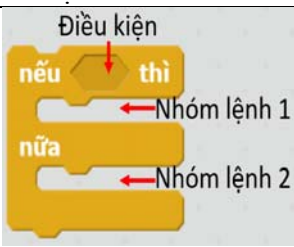


Hình 2: Minh họa câu lệnh điều kiện với A - B

Trong bài toán tính hiệu 2 số A và B của chúng ta, câu lệnh điều kiện của chúng ta có thể minh họa như Hình 2. Bây giờ, sau khi người dùng nhập vào 2 số A và B, chúng ta sẽ kiểm tra A có thực sự lớn hơn B hay không. Nếu kết quả là đúng ($A > B$), chúng ta mới thực hiện phép trừ. Ngược lại (kết quả là sai), chúng ta sẽ thông báo lên là chương trình không hợp lệ. Như vậy, với một cặp giá trị

A và B, chỉ 1 trong 2 nhánh trên được thực hiện. Sẽ không bao giờ cả 2 nhánh được thực hiện trong câu lệnh điều kiện




Câu lệnh điều kiện trong Scratch được nằm trong nhóm lệnh Điều khiển, có màu vàng:


Câu lệnh	Chức năng
	<p>Câu lệnh này có 3 phần:</p> <p>Phần hình thoi: Đây là nơi chúng ta đặt điều kiện cho câu lệnh. Để thực hiện một điều kiện nào đó, chúng ta cần sử dụng các câu lệnh trong nhóm Phép toán, có màu xanh lá cây. Xem hướng dẫn chi tiết ở phần tiếp theo</p> <p>Phần nhóm câu lệnh thứ nhất: Một hoặc nhiều câu lệnh sẽ được đặt ngay dưới câu lệnh điều kiện. Khi điều kiện (phần hình thoi) là đúng, thì nhóm câu lệnh này sẽ được thực hiện</p> <p>Phần nhóm câu lệnh thứ hai: Khi điều kiện là sai, nhóm câu lệnh này sẽ được thực hiện. Nhóm câu lệnh này được ghép nối trong phần nữa.</p>

3. Xây dựng điều kiện

(10 phút)

Để xây dựng một điều kiện, chúng ta sẽ sử dụng các câu lệnh thuộc nhóm phép toán và có hình thoi thì mới có thể ghép vào câu lệnh điều kiện. Một số câu lệnh điều kiện thông dụng:

Câu lệnh	Chức năng
	So sánh nhỏ hơn: Nếu phần tử thứ nhất nhỏ hơn phần tử thứ 2, thì điều kiện là đúng.
	So sánh lớn hơn: Nếu phần tử thứ nhất lớn hơn phần tử thứ 2, thì điều kiện là đúng.
	So sánh bằng: Nếu 2 phần tử bằng nhau, thì điều kiện là đúng

Một lưu ý quan trọng đối với việc xây dựng điều kiện, là bên cạnh việc nhập vào một con số nhất định cho các ô màu trắng, chúng ta cũng có thể kéo thả các biến số (A hoặc B hoặc Kết Quả) vào các ô này. Trong yêu cầu của bài học này, chúng ta cần xây dựng điều kiện A lớn hơn B. Do đó, chúng ta sẽ sử dụng câu lệnh thứ 2, , và kéo thả biến A vào phần tử thứ nhất, biến B vào phần tử thứ 2, như minh họa ở Hình 3.



Hình 3: Tạo điều kiện $A > B$

Từ đây, chúng ta có thể kéo thả khối điều kiện này vào phần hình thoi của câu lệnh điều kiện.

4. Bài tập tự chọn trên lớp theo nhóm – Lưu kết quả của học sinh

(15 phút)

Giáo viên cho học sinh làm bài tập tự chọn bất kỳ trong chương trình đã giảng dạy.

Ví dụ: Giáo viên cho học sinh hoàn thiện chương trình tính hiệu 2 số A và B với yêu cầu chỉ tính hiệu $A - B$ khi A lớn hơn B. Ngược lại, hiện câu thông báo không hợp lệ ra màn hình.

Chương trình gợi ý như sau:

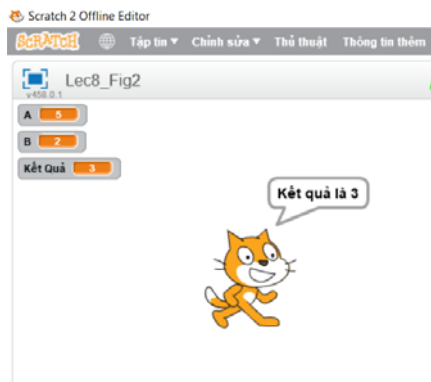
```

khi nhấn vào
hỏi Nhập số A và đợi
đặt A là trả lời
hỏi Nhập số B và đợi
đặt B là trả lời
nếu A < B thì
nói Không hợp lệ trong 2 giây
nếu
đặt Kết Quả là A - B
nói nói Kết quả là và Kết Quả trong 2 giây

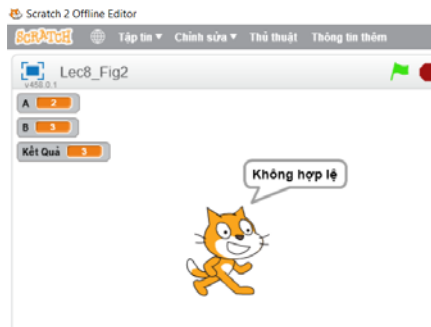
```

Hình 4: Chương trình tính hiệu A và B hoàn chỉnh

Khi chạy chương trình này, sẽ có 2 thông báo hiển thị ra màn hình, tùy vào các giá trị của A và B.



Hình 5: Thông báo khi $A = 5, B = 2$



Hình 6: Thông báo khi $A = 2, B = 3$