

NGUYỄN HỮU HƯNG (Chủ biên)

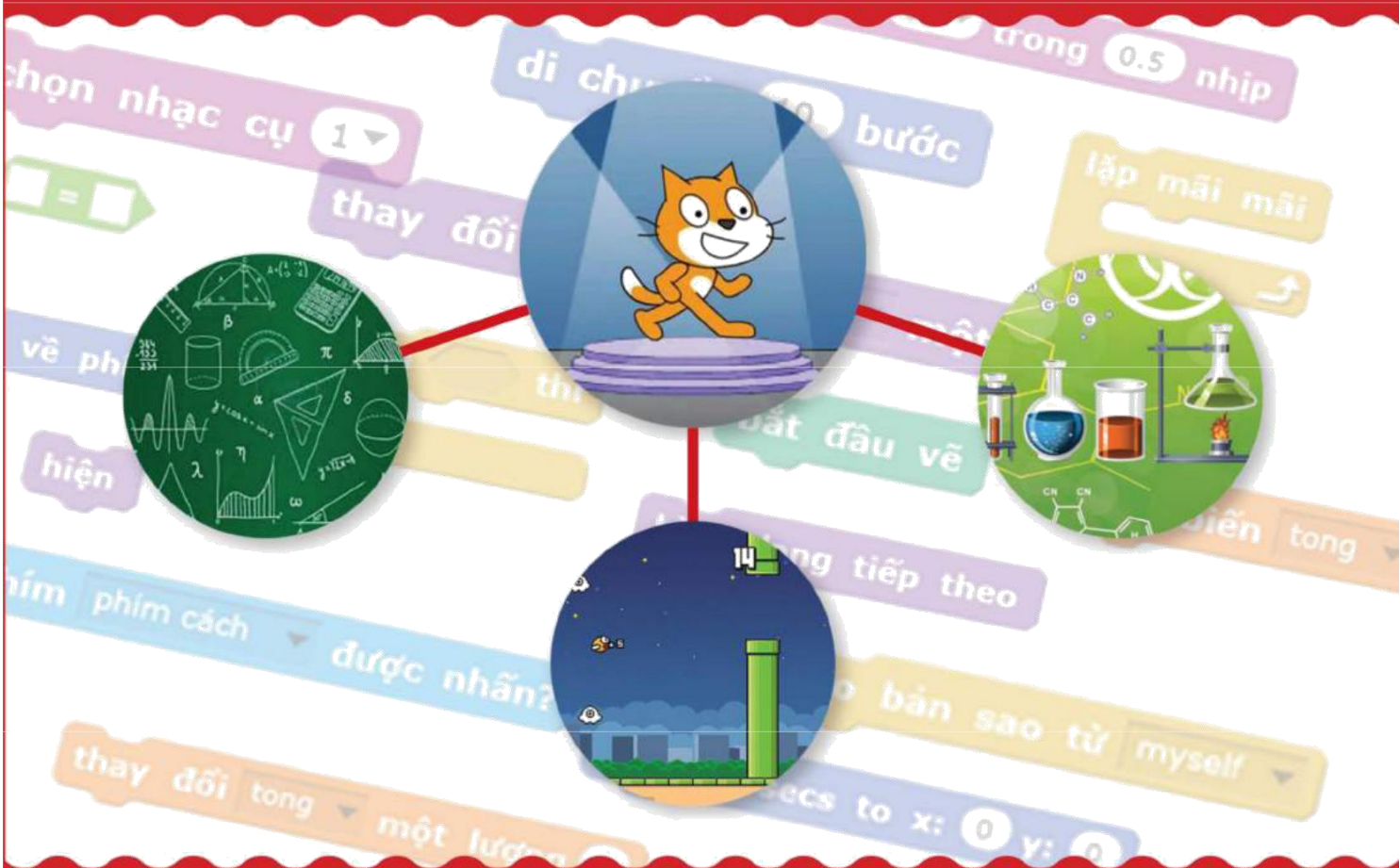
TẠ THỊ BÍCH – DƯƠNG LỰC – NGUYỄN THỊ MINH

LẬP TRÌNH

với

SCRATCH

HÀNH TRANG CHO TƯƠNG LAI



NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

LỜI NÓI ĐẦU

Học lập trình ở Việt Nam

Từ năm học 2006 – 2007, học sinh đã được tiếp cận với máy tính thông qua môn Tin học từ lớp 3 tới lớp 12. Tuy nhiên, với định hướng chương trình và đặc thù của các nhà trường do vậy lập trình chỉ là một phần nhỏ được giới thiệu trong toàn bộ chương trình.

Học lập trình trên thế giới

Bill Gates lập trình phần mềm Tic-tac-toe khi 13 tuổi. Zuckerberg đã bắt đầu lập trình máy tính từ lúc học lớp 6. Elon Musk đã tự mình học lập trình khi còn rất nhỏ, và ở tuổi 12 cậu đã bán phần mềm đầu tiên của mình – một trò chơi không gian tên là Blastar – với giá khoảng 500 đô la.

Từ năm 2000, lập trình là môn học bắt buộc tại các trường trung học ở Israel.

Năm 2014, chính phủ Anh công bố chương trình dạy lập trình bắt buộc cho học sinh từ 5 tuổi.

Đầu tháng 6/2016, chính phủ Nhật đã chính thức thông qua chương trình giáo dục **lập trình mới** cho học sinh phổ thông. Theo đó, tất cả các học sinh cấp 2 trường công tại Nhật sẽ bắt buộc phải học lập trình ngay từ đầu cấp. Mục tiêu của chương trình này là để chuẩn bị lực lượng nhân sự sẵn sàng cho cái được chính phủ gọi là **“Cuộc cách mạng công nghiệp lần thứ 4”**, đó sẽ là kỉ nguyên phát triển bùng nổ của nhiều ngành như robot và trí tuệ nhân tạo, kết hợp giữa các hệ thống ảo và thực thể, vạn vật kết nối Internet (IoT)...

Mục đích của cuốn sách

Giới thiệu về SCRATCH, một môi trường và ngôn ngữ lập trình hoàn toàn mới dành riêng cho học sinh phổ thông, rất đơn giản, trực quan, đang ngày càng phổ biến rộng rãi trên thế giới. Thông qua SCRATCH học sinh có thể thỏa sức sáng tạo và chia sẻ các sản phẩm công nghệ thông tin như: Thiệp, phim hoạt hình, ứng dụng, trò chơi...



Yêu cầu kiến thức khi sử dụng sách

Để sử dụng sách hiệu quả, học sinh cần có một số kiến thức nhất định về sử dụng chuột và bàn phím máy tính, biết đếm đến 480, hiểu được phép nhân và phép chia. Nhiều khái niệm phức tạp hơn như số âm, phân số, trục tọa độ, số đo góc sẽ được giải thích bổ sung trong nội dung cuốn sách.

Đối tượng sử dụng sách

Học sinh Tiểu học (từ 9 đến 10 tuổi có sự hỗ trợ của phụ huynh và Trung học cơ sở).

Cảm ơn

Chúng tôi muốn cảm ơn những người sáng tạo ra Scratch. Nhóm Lifelong Kindergarten tại MIT Media Lab (Massachusetts Institute of Technology) do Giáo sư Mitch Resnick đứng đầu đã tạo ra một cách thức tuyệt vời để dạy học sinh học lập trình nền tảng cơ bản và làm chủ công nghệ trong kỉ nguyên mới. Trang chủ: <https://scratch.mit.edu>

Chúng tôi cũng muốn cảm ơn các thành viên nhóm tác giả của tôi đã miệt mài không quản ngày đêm với mong muốn nhanh chóng đưa SCRATCH phổ biến cho cộng đồng học sinh Việt Nam.

Để có thêm thông tin hỗ trợ, tài liệu và bản mềm các ví dụ trong sách, các bạn có thể truy cập website: Daylaptrinh.net (Link: <http://daylaptrinh.net/post/tro-giup/tai-lieu-cho-sach-in.html>).



MỤC LỤC

Lời nói đầu	3
Chương I. Phần mềm Scratch 2.0 Offline Editor	8
Bài 1. Giao diện.....	10
1.1. Khu vực sân khấu	10
1.2. Khu vực quản lí hình nền và nhân vật.....	11
1.3. Thanh danh mục, Thanh công cụ và Cửa sổ trợ giúp	14
Bài 2. Thanh danh mục và thanh công cụ.....	16
2.1. Thanh danh mục.....	16
2.2. Thanh công cụ.....	19
Bài 3. Sân khấu	23
3.1. Hệ toạ độ trong sân khấu.....	23
3.2. Các c ế độ của s n khấu	24
Bài 4. Làm việc với hình nền và ngoại hình nhân vật	27
4.1. Chọn từ thư viện.....	27
4.2. Vẽ mới.....	28
4.3. Tải từ máy tính	28
4.4. Chụp ảnh.....	29
Bài 5. Sử dụng công cụ vẽ	31
5.1. Chế độ Bitmap.....	31
5.2. Chế độ Vector.....	34
Bài 6. Khu âm thanh	37
6.1. Thêm và xoá âm thanh	37
6.2. Chỉnh sửa và thêm hiệu ứng cho âm thanh	39
Bài 7. Cửa sổ trợ giúp	41
7.1. Step-by-Step.....	41
7.2. How To	42
7.3. Blocks.....	43
Bài 8. Khu lập trình	44
8.1. Bảng danh mục các khối lệnh.....	44



Bài 9. Ví dụ cơ bản	49
9.1. Thêm mới nhân vật	49
9.2. Thay đổi tâm nhân vật	49
9.3. Vẽ cạnh đầu tiên	50
9.4. Hoàn thành hình vuông	51
Chương II. Thiệp sinh nhật	53
Tổng quan chương trình	53
Bài 1. Tạo hình nền và vẽ chong chóng	55
1.1. Hình nền	55
1.2. Chong chóng	56
Bài 2. Lập trình chong chóng quay	60
2.1. Khối lệnh lặp và xoay	60
2.2. Điều chỉnh tâm của nhân vật	61
2.3. Đối tượng que	62
Bài 3. Lập trình đối tượng Chữ	65
3.1. Thêm và chỉnh sửa chữ	65
3.2. Hiệu ứng	66
Bài 4. Lập trình đối tượng Bóng bay	69
4.1. Thêm nhân vật Bóng bay	69
4.2. Lập trình bóng bay	70
Bài 5. Vị trí bắt đầu của các đối tượng	74
5.1. Nhảy tới vị trí nhất định	74
Bài 6. Bóng bay bay lên từ vị trí ngẫu nhiên	77
6.1. Vòng lặp lồng nhau	77
6.2. Lấy giá trị ngẫu nhiên	77
Bài 7. Lập trình đối tượng nến	80
7.1. Thêm nhân vật từ thư mục	80
7.2. Lập trình cho đối tượng nến	81
Bài 8. Đối tượng Mũ và Pháo	83
8.1. Thêm nhân vật từ thư mục	83
8.2. Lập trình cho Mũ	83
8.3. Lập trình cho Pháo	85
Bài 9. Âm thanh	88
9.1. Thêm âm thanh	88
9.2. Lập trình	89
Bài 10. Lập trình với bản sao	91
10.1. Thêm đối tượng Ngôi sao	91
10.2. Tạo bản sao	91
10.3. Lập trình thay đổi hiệu ứng cho bản sao	92
10.4. Sắp xếp các bản sao của Ngôi sao	94



Chương III. Đồng hồ	98
Tổng quan chương trình	98
Bài 1. Vẽ Mặt đồng hồ	100
1.1. Vẽ mặt đồng hồ	100
Bài 2. Giới thiệu về góc và hướng của đối tượng	104
2.1. Giới thiệu về góc	104
2.2. Hướng của đối tượng	104
Bài 3. Vẽ kim đồng hồ	107
3.1. Kim giây	107
3.2. Kim phút	107
3.3. Kim giờ	108
3.4. Chỉnh lại vị trí các số	108
Bài 4. Lập trình kim giây	112
4.1. Hướng của đối tượng	112
4.2. Lập trình kim giây	112
Bài 5. Lập trình kim phút	115
5.1. Lập trình kim phút	115
Bài 6. Lập trình kim giờ	116
6.1. Lập trình kim giờ	116
Chương IV. Đường lên đỉnh Olympia	119
Tổng quan chương trình	119
Bài 1. Vẽ đường leo núi	121
1.1. Vẽ đường leo núi	121
1.2. Vẽ bậc	123
Bài 2. Vẽ núi và mây	125
2.1. Vẽ núi	125
2.2. Vẽ mây	128
Bài 3. Lập trình cho mèo máy nói và nghĩ	129
3.1. Lập trình để đối tượng nói	129
3.2. Lập trình để đối tượng nghĩ	130
3.3. Lập trình cho Doraemon	130
Bài 4. Giới thiệu về biến	132
4.1. Biến trong Scratch	132
Bài 5. Lập trình cho Doraemon tính toán	136
5.1. Tính tổng hai số ngẫu nhiên	136
Bài 6. Lập trình để Doraemon hỏi	138
6.1. Khối lệnh hỏi	138

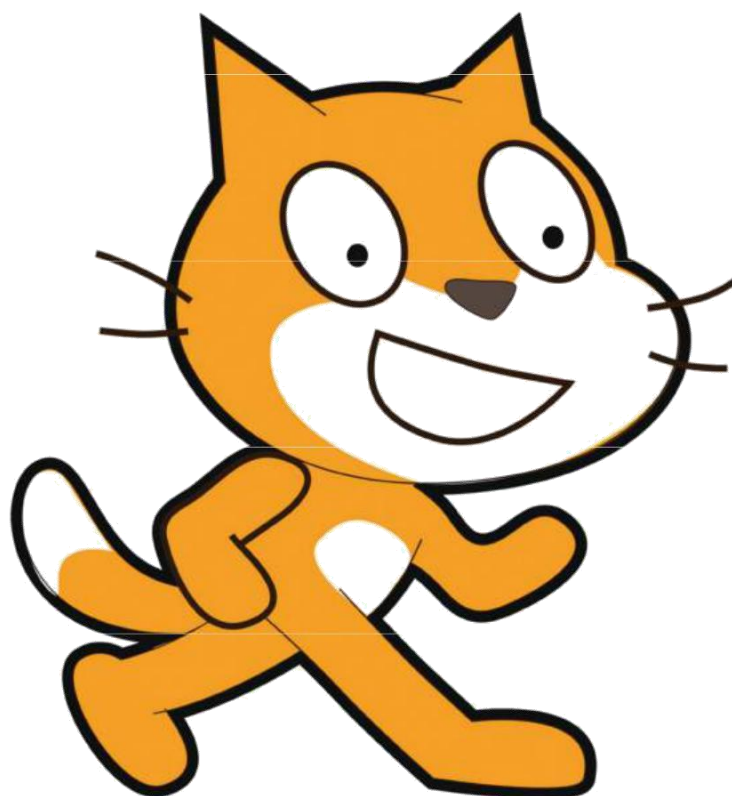


Bài 7. Lập trình để Doraemon kiểm tra câu trả lời	142
7.1. Hai khối lệnh kiểm tra điều kiện	142
7.2. Kiểm tra câu trả lời	143
Bài 8. Lập trình cho Nobita	145
8.1. Khối lệnh Glide	145
Bài 9. Kết thúc chương trình	148
9.1. Hoàn thiện chương trình	148
Chương V. Bảo vệ đảo	150
Tổng quan chương trình	150
Bài 1. Vẽ bản đồ	152
1.1. Vẽ đất liền	152
1.2. Vẽ hai đảo	155
Bài 2. Vẽ Pháo và Tàu địch	158
2.1. Vẽ Pháo	158
2.2. Vẽ Tàu địch	159
Bài 3. Lập trình đối tượng Pháo và Tâm ngắm	161
3.1. Lập trình đối tượng Pháo	161
3.2. Lập trình đối tượng Tâm ngắm	162
Bài 4. Lập trình cho Lửa	165
4.1. Lập trình vị trí của Lửa	165
4.2. Lập trình đối tượng Lửa chỉ hiện khi nhấn chuột	167
Bài 5. Lập trình cho Tàu địch	170
5.1. Vị trí của tàu khi hướng về Đảo 1	170
5.2. Vị trí của tàu khi hướng về Đảo 2	172
Bài 6. Lập trình cho Tàu địch (tiếp theo)	174
6.1. Lập trình cho bản sao	174
Bài 7. Lập trình cho Tàu địch (tiếp theo)	178
7.1. Lập trình cho Tàu bị bắn	178
Bài 8. Kết thúc chương trình	181
8.1. Thêm và lập trình đối tượng chữ Start	181
8.2. Thêm và lập trình đối tượng chữ Game Over	182
Phụ lục: Mô phỏng mạch điện	186
Tổng quan chương trình	186
Thiết kế các đối tượng	188
Lập trình	188



Phần mềm Scratch 2.0 Offline Editor

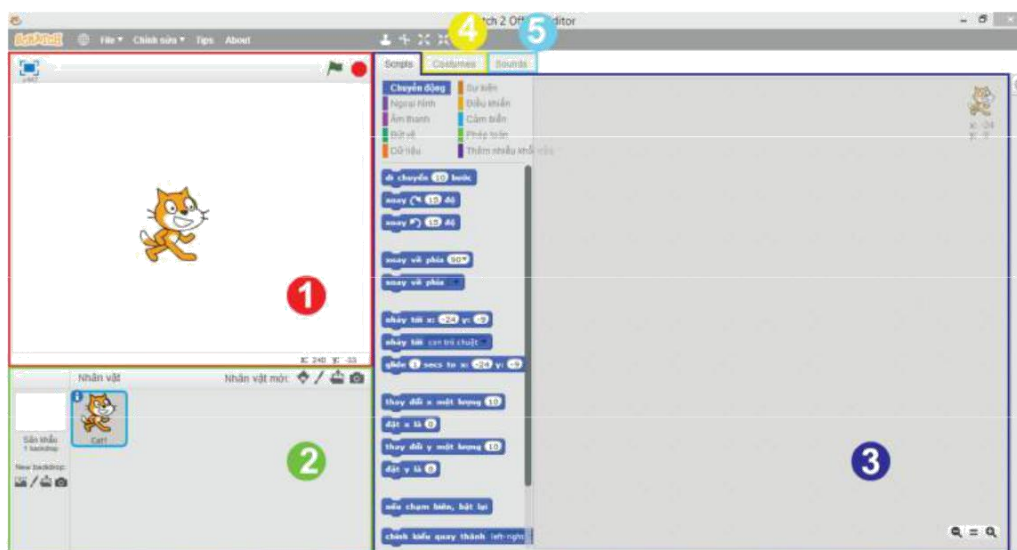
Trong chương này, các bạn sẽ được tìm hiểu về phần mềm Scratch 2.0 Offline Editor, các thành phần của giao diện và các chức năng của từng phần. Sau khi kết thúc chương, các bạn sẽ có được cái nhìn tổng quan và cách sử dụng một số chức năng cơ bản của phần mềm.



BÀI 1

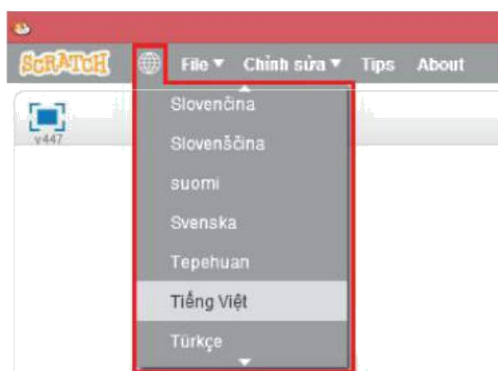
GIAO DIỆN

Giao diện ngầm định của Scratch.



Hình 1.1.1. Giao diện phần mềm Scratch

Ngôn ngữ hiển thị ngầm định của chương trình là tiếng Anh. Các bạn có thể chuyển đổi giao diện sang bản Việt hoá bằng cách nháy chuột vào biểu tượng quả địa cầu, di chuyển chuột xuống dưới và chọn Tiếng Việt, ngôn ngữ hiển thị giao diện hầu hết sẽ là tiếng Việt.



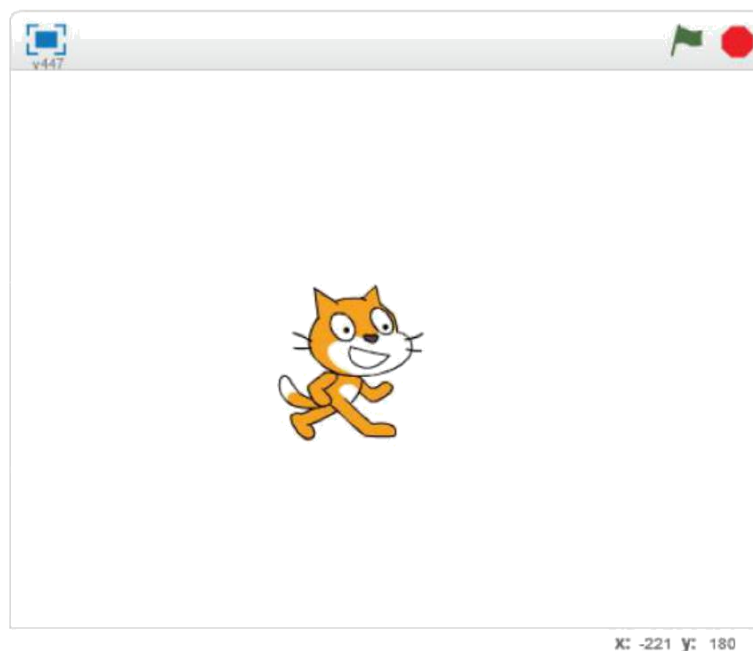
Hình 1.1.2. Thay đổi ngôn ngữ hiển thị trên giao diện

Giao diện Scratch có năm khu vực chính như mô tả trên hình 1.1.1.

1.1. Khu vực sân khấu

Khu vực sân khấu (được đánh số 1) là nơi biểu diễn từng phần hoặc cả dự án. Khi bạn khởi động phần mềm, một chú mèo xuất hiện trên sân khấu – đó là nhân vật ngầm định của chương trình.

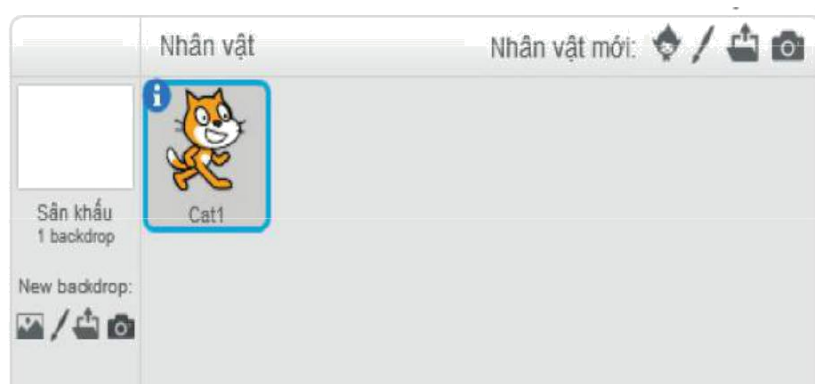




Hình 1.1.3. Khu vực sân khấu

1.2. Khu vực quản lí phong nền sân khấu và nhân vật

Khu vực này gồm các biểu tượng thu nhỏ gồm khung chứa phong nền sân khấu và khung nhân vật (gồm danh sách các nhân vật hoặc đối tượng) trong chương trình.



Hình 1.1.4. Khu vực quản lí (phông) nền sân khấu và nhân vật

1.2.1. Nhân vật

Trong khung nhân vật có thể có nhiều nhân vật. Mỗi nhân vật sẽ có ba khu tác động vào, đó là Khu lập trình ③, Khu ngoại hình ④ và Khu âm thanh ⑤.

a) Khu lập trình

Hiện ra khi chọn trang **Scripts**, là nơi các khối lệnh được ghép với nhau tạo thành một chương trình theo ý muốn của lập trình viên.

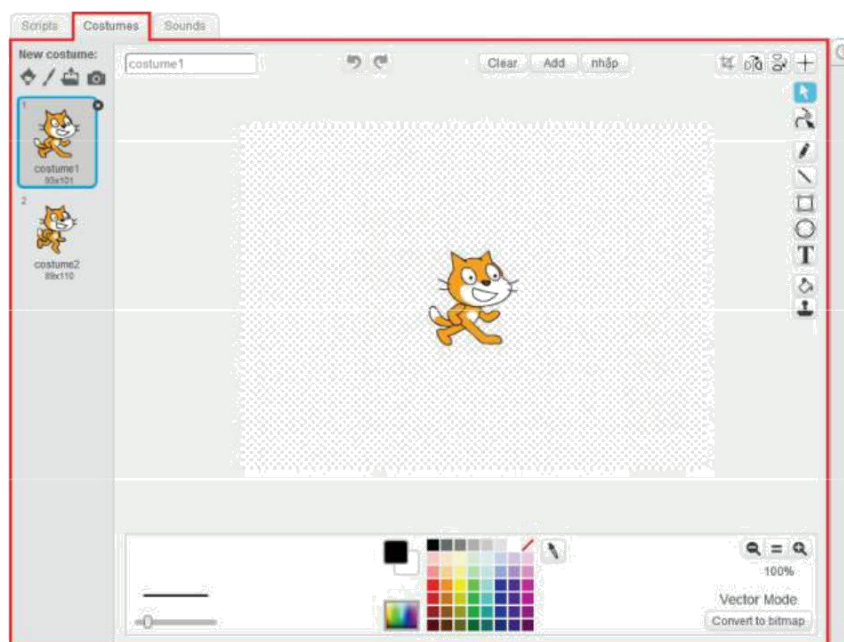




Hình 1.1.5. Khu lập trình

b) Khu ngoại hình

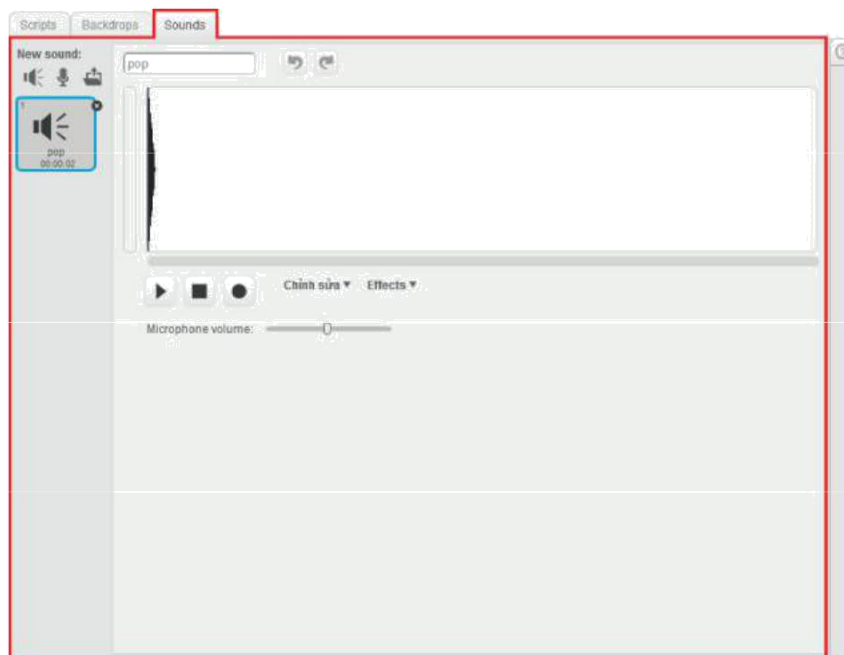
Mỗi nhân vật sẽ có một hoặc nhiều hình ảnh hiển thị. Khu ngoại hình sẽ xuất hiện khi chọn trang **Costume**, cho phép các lập trình viên thao tác với hình ảnh của nhân vật như thêm mới, chỉnh sửa, xoá các hình ảnh trên cơ sở thư viện có sẵn hoặc tự tạo theo yêu cầu.



Hình 1.1.6. Khu ngoại hình

c) Khu âm thanh

Xuất hiện khi chọn trang **Sounds**, cho phép lập trình viên tạo, chỉnh sửa âm thanh cho các nhân vật trên cơ sở thư viện có sẵn hoặc tự tạo theo yêu cầu.



Hình 1.1.7. Khu âm thanh

1.2.2. Phòng nền sân khấu

Phòng nền sân khấu có thể coi là một đối tượng (nhân vật) đặc biệt, cũng có ba khu tác động vào, đó là Khu lập trình, Khu phòng nền và Khu âm thanh. Về cơ bản, 3 khu này có thành phần và chức năng tương đương với ba khu tác động vào nhân vật bình thường, chỉ có một chút khác biệt:

a) Khu lập trình

Giống với Khu lập trình của nhân vật, tuy nhiên vì sân khấu không thể di chuyển nên số lượng các khối lệnh sẽ ít hơn.

b) Khu phòng nền

Khu phòng nền của sân khấu xuất hiện khi chọn trang **Backdrops**, là nơi quản lý các phòng nền (Khu này cũng chính là Khu ngoại hình của nhân vật).

c) Khu âm thanh

Giống hoàn toàn với khu âm thanh của nhân vật.

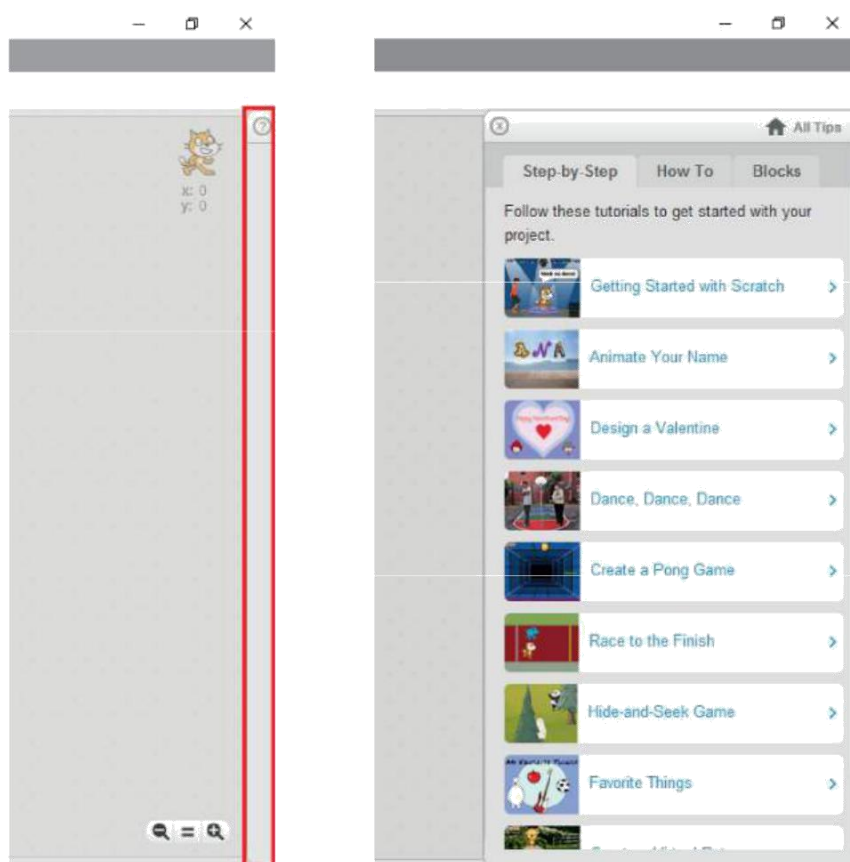


1.3. Thanh danh mục, Thanh công cụ và Cửa sổ trợ giúp

Ngoài 5 khu vực chính, giao diện phần mềm Scratch còn có Thanh danh mục, Thanh công cụ và Cửa sổ trợ giúp mà sẽ được nhắc đến trong những bài sau.



Hình 1.1.8. Thanh danh mục và Thanh công cụ



Hình 1.1.9. Cửa sổ trợ giúp



Tóm tắt lý thuyết và bài tập thực hành

Giao diện - Interface

Cấu tạo giao diện Scratch



daylaptrinh.net

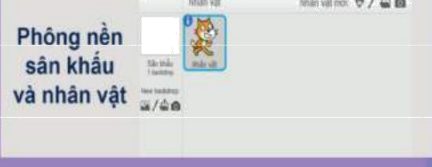
Ngôn ngữ Thanh danh mục Thanh công cụ



Sân khấu nhân vật



Phòng nền sân khấu và nhân vật



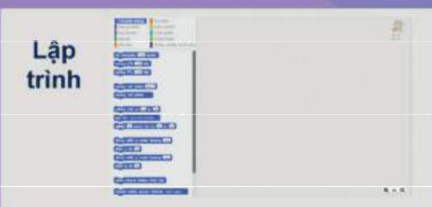
Giao diện - Interface

Cấu tạo giao diện Scratch

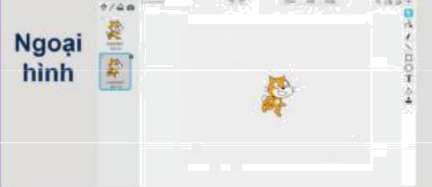


daylaptrinh.net

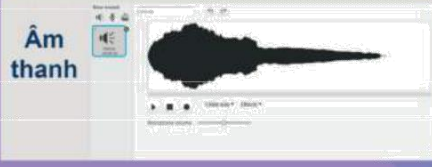
Lập trình



Ngoại hình



Âm thanh





BÀI 2

THANH DANH MỤC VÀ THANH CÔNG CỤ

2.1. Thanh danh mục

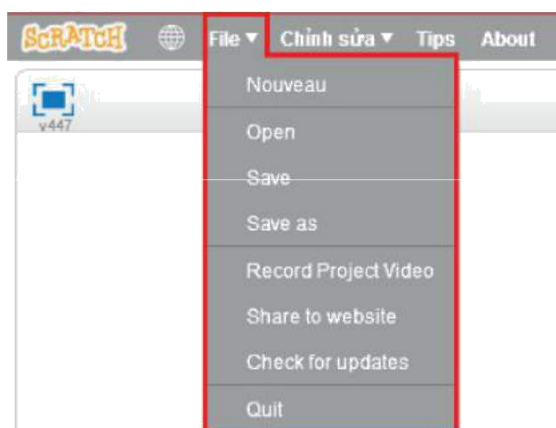


Hình 1.2.1. Thanh danh mục

Thanh danh mục nằm ở phía trên cùng của giao diện và có vai trò rất quan trọng liên quan tới thao tác sử dụng chương trình Scratch. Sau đây là từng thành phần của Thanh danh mục:

1. File

Đây là nơi thao tác với tập tin dự án. Tập tin là một tập hợp các thông tin được lưu lại trên bộ nhớ máy tính. Ví dụ một bức ảnh, một bài hát, một bộ phim,... lưu lại trên bộ nhớ máy tính, chúng ta đều gọi đó là tập tin. Mỗi dự án Scratch cũng là một tập tin.



Hình 1.2.2. File

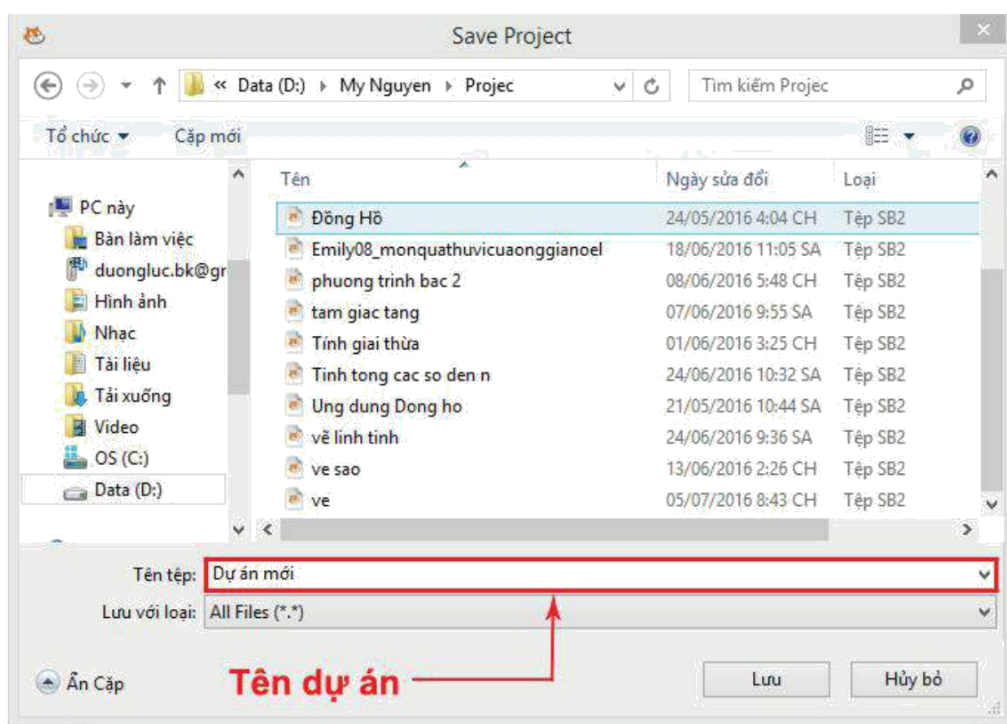
- Lựa chọn **New**: Do bản Việt hoá chưa được hoàn thiện ở phiên bản v445 nên có chút nhầm lẫn thành Nouveau. Lựa chọn này sẽ tạo một dự án mới để làm việc. Khi các bạn đang làm việc với một dự án khác mà chưa lưu lại, nếu chọn **New** thì sẽ có một cửa sổ hiện ra hỏi bạn có muốn lưu lại hay không.



Hình 1.2.3. Cửa sổ xác nhận lưu dự án

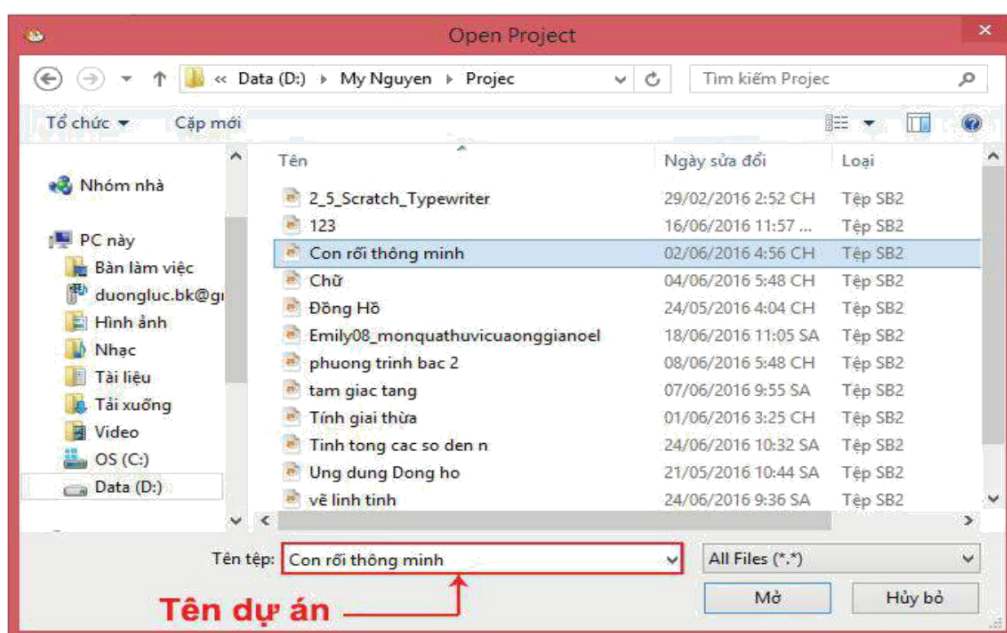


- Nếu chọn **Save** (Lưu dự án) thì một cửa sổ khác sẽ hiện ra để các bạn đặt tên và lựa chọn vị trí lưu lại trên máy tính.



Hình 1.2. . Cửa sổ Lưu dự án

- Lựa chọn **Open**: Dùng để mở một dự án đã có sẵn trên máy tính, một cửa sổ sẽ hiện ra để các bạn chọn tập tin dự án trên máy tính.



Hình 1.2.5. Cửa sổ mở dự án



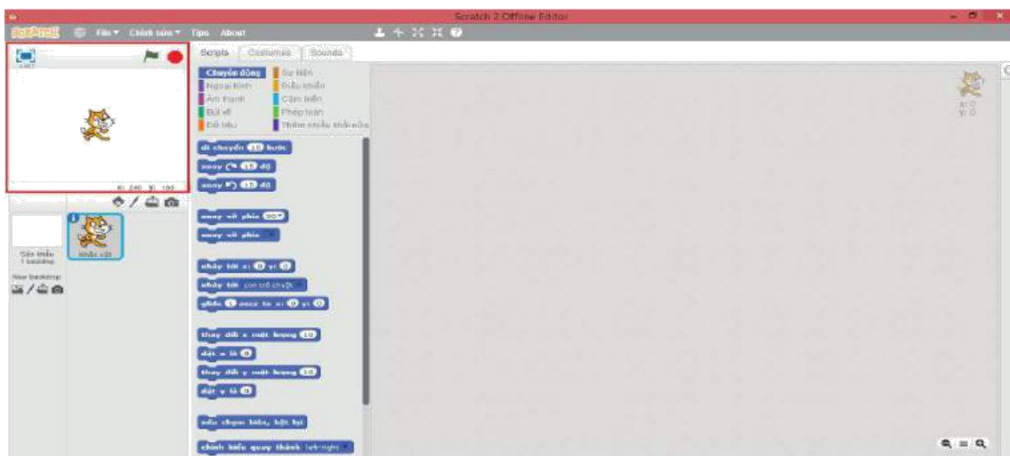
- Lựa chọn **Save**: Dùng để lưu lại dự án đang làm việc. Khi chọn **Save**, giao diện sẽ xuất hiện một cửa sổ như trong *Hình 1.2.4* cho phép đặt tên và lựa chọn vị trí lưu lại trên máy tính. Cửa sổ này chỉ xuất hiện khi các bạn làm việc với dự án mới và nháy **Save** lần đầu. Với những lần nhấn **Save** tiếp theo hoặc nếu các bạn mở một dự án có sẵn bằng **Open**, cửa sổ sẽ không hiện ra vì chương trình đã biết tên và vị trí của tập tin dự án.
- Lựa chọn **Save as**: Lưu lại tập tin dự án dưới dạng bản sao. Chương trình sẽ lưu dự án thành một tập tin mới trên máy tính. Cửa sổ lưu dự án như trong *Hình 1.2.4* sẽ luôn xuất hiện để các bạn đặt tên và chọn vị trí lưu cho tập tin mới.
- Lựa chọn **Record Project Video**: ghi hình lại dự án thành video.
- Lựa chọn **Share to website**: chia sẻ dự án lên website.
- Lựa chọn **Check for updates**: Kiểm tra cập nhật phiên bản cho chương trình.
- Lựa chọn **Quit**: Thoát khỏi chương trình. Nếu các bạn chưa lưu dự án thì cửa sổ xác nhận như trong *Hình 1.2.3* sẽ xuất hiện. Sau khi tiến hành lưu (hoặc sau khi các bạn chọn không lưu), chương trình sẽ thoát.

2. Chỉnh sửa



Hình 1.2.6. Chỉnh sửa

- Lựa chọn **Xoá**: Lấy lại đối tượng hoặc khối lệnh vừa bị xoá đi. Tiếng Anh là **Undo**, bản Việt hoá có chút sai sót nên dịch là **Xoá**. Lựa chọn này chỉ lấy lại được đối tượng hoặc khối lệnh bị xoá gần nhất.
- Lựa chọn **Small stage layout**: Chuyển sân khấu từ chế độ có kích thước mặc định sang chế độ thu nhỏ và ngược lại.

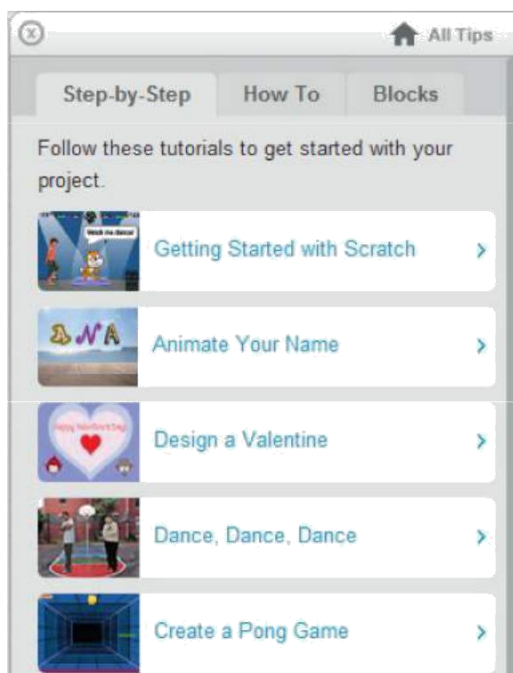


Hình 1.2.7. Chế độ thu nhỏ của sân khấu

- Lựa chọn **Turbo mode**: Bật/Tắt chế độ Turbo. Chế độ này giúp chương trình tăng tốc độ xử lý các khối lệnh.

3. Tips

Khi chọn mục này, cửa sổ trợ giúp sẽ hiện ra, chứa những dự án mẫu và hướng dẫn giải thích về các khối lệnh.

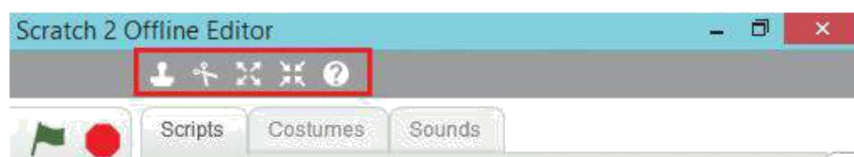


Hình 1.2.8. Cửa sổ trợ giúp

4. About

Mục này chứa đường dẫn đến trang web giới thiệu về Scratch.

2.2. Thanh công cụ



Hình 1.2.9. Thanh công cụ

Thanh công cụ nằm ở bên phải Thanh danh mục, chứa các công cụ giúp thao tác với đối tượng và khối lệnh nhanh hơn. Khi các bạn chọn công cụ nào, con trỏ chuột sẽ thay đổi hình dạng theo công cụ đó. Khi muốn huỷ không sử dụng công cụ đó nữa, các bạn chỉ cần nháy chuột vào một chỗ trống trên sân khấu, con trỏ chuột sẽ trở lại như hình dáng ban đầu. Thanh công cụ gồm các thành phần:



1. Tạo bản sao

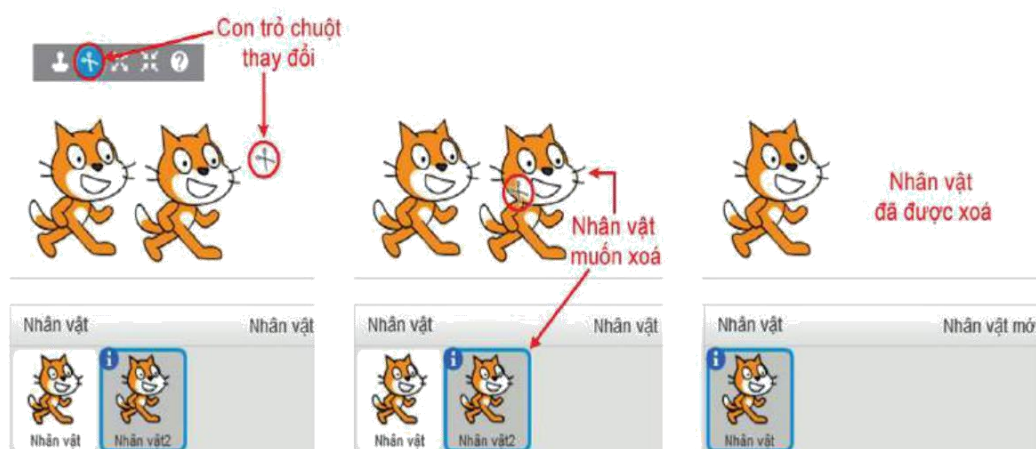
Công cụ này giúp các bạn tạo bản sao từ đối tượng hoặc khối lệnh. Các bạn chỉ cần nháy chuột chọn công cụ **Tạo bản sao**, sau đó chọn đối tượng hoặc đoạn khối lệnh muốn sao chép rồi kéo thả tới vị trí khác.



Hình 1.2.10. Tạo bản sao

2. Xoá

Công cụ này giúp các bạn xoá đối tượng hoặc khối lệnh. Các bạn chỉ cần nháy chuột chọn công cụ , sau đó nháy chuột vào đối tượng hay khối lệnh muốn xoá.



Hình 1.2.11. Xoá

Lưu ý: Khi xoá đối tượng thì tất cả khối lệnh tác động lên đối tượng cũng bị xoá.

3. Grow

Công cụ này giúp phóng to đối tượng lên. Cách sử dụng cũng giống như công cụ **Xoá**: Chọn công cụ sau đó nháy chuột vào đối tượng muốn phóng to nhiều lần cho đến khi đạt kích thước như ý (Hình 1.2.12).

4. Shrink

Ngược lại với công cụ  là công cụ  với cách sử dụng tương tự.



Hình 1.2.12: Phóng to và thu nhỏ đối tượng

5. Trợ giúp

Khi các bạn không hiểu và cần hỗ trợ về khối lệnh nào, nháy chuột vào biểu tượng  sau đó chọn khối lệnh mong muốn, cửa sổ trợ giúp sẽ xuất hiện, chứa hướng dẫn sử dụng khối lệnh đó và ví dụ mẫu.



Hình 1.2.13. Trợ giúp



Tóm tắt lí thuyết và bài tập thực hành

Danh mục - Menu

Cách sử dụng thanh danh mục

File ▾ | **Chỉnh sửa ▾** | Tips | About

File ▾

- Nouveau
- Open
- Save
- Save as
- Record Project Video
- Share to website
- Check for updates
- Quit

Chỉnh sửa ▾

- xoá
- Small stage layout
- Turbo mode

daylaptrinh.net

File

- Nouveau → Tạo mới
- Open → Mở
- Save → Lưu
- Save as → Lưu dạng bản sao
- Record Project Video → Ghi hình
- Share to website → Chia sẻ
- Check for updates → Cập nhật
- Quit → Thoát

Chỉnh sửa

- xoá → Lấy lại
- Small stage layout → Sân khấu nhỏ
- Turbo mode → Chế độ nhanh

Trợ giúp → **Tips** | About → Giới thiệu Scratch

Công cụ - Tools

Cách sử dụng thanh công cụ

+ =

+ =

+ =

+ =

daylaptrinh.net

Tạo bản sao

Tạo ra một bản sao của nhân vật hoặc khối lệnh

Xoá

Xoá một nhân vật hoặc khối lệnh

Phóng to

Phóng to nhân vật

Thu nhỏ

Thu nhỏ nhân vật

Trợ giúp

Cửa sổ trợ giúp

BÀI 3

SÂN KHẤU

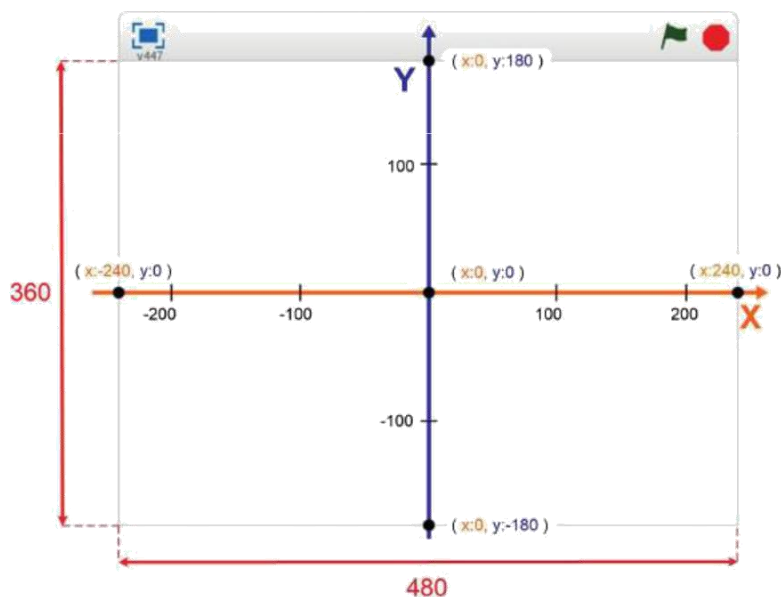
Sân khấu là nơi biểu diễn các hành động, hình ảnh, âm thanh của các đối tượng khi chạy chương trình, là nơi hiển thị kết quả của dự án.



Hình 1.3.1. Khu vực sân khấu.

3.1. Hệ tọa độ trong sân khấu

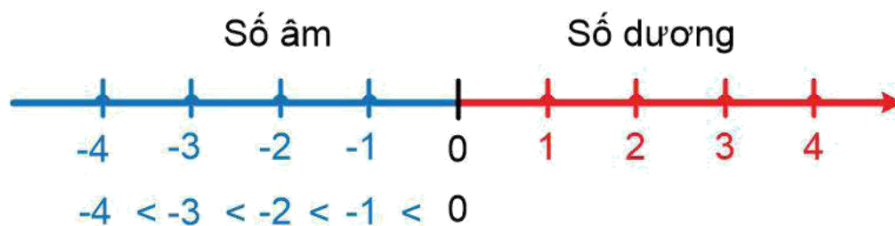
Sân khấu có chiều rộng là 360 và chiều dài là 480, được xác định theo hệ trục tọa độ xOy trong toán học. Trục x theo chiều ngang có giá trị từ -240 đến 240 , trục y theo chiều dọc có giá trị từ -180 đến 180 .



Hình 1.3.2. Hệ tọa độ trong sân khấu

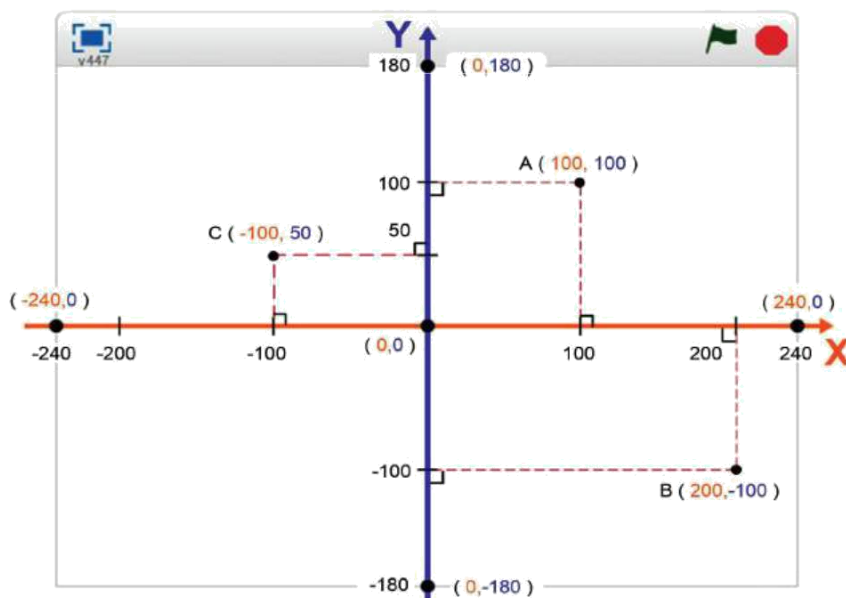


Trong hệ tọa độ có sử dụng đến số âm. Số âm là số nhỏ hơn 0, được biểu diễn bằng dấu trừ (-) đứng trước số dương tương ứng. Trên trục số thì số âm đứng bên trái số 0.



Hình 1.3.3. Trục số

Toạ độ của một điểm được xác định bởi vị trí theo trục x và vị trí theo trục y (gọi tắt là vị trí x và vị trí y) của điểm đó. Để xác định toạ độ của một điểm trên sân khấu, từ điểm đó ta kẻ một đường thẳng vuông góc với trục x , đường thẳng đó cắt trục x tại điểm có x bằng bao nhiêu thì đó sẽ là vị trí x của điểm đó. Tương tự nh với trục y , ta tìm được vị trí y . Cặp số (x,y) chính là toạ độ của điểm đó. Trục x và trục y giao nhau tại điểm có toạ độ $(0,0)$.



Hình 1.3.4. Xác định toạ độ của một điểm

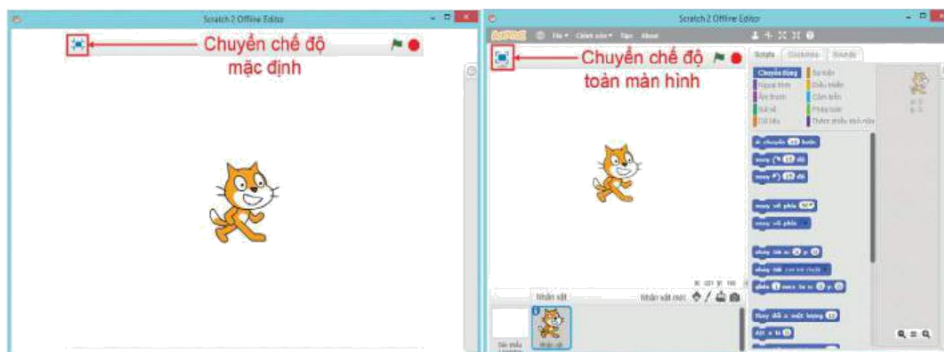
3.2. Các chế độ của sân khấu

Sân khấu có ba chế độ: **chế độ mặc định**, **chế độ thu nhỏ** và **chế độ toàn màn hình**.

1. **Chế độ mặc định**: là chế độ thiết đặt sẵn khi mở chương trình.

2. **Chế độ thu nhỏ**: được minh hoạ như trong Hình 1.2.7. Ở chế độ này, chiều dài và chiều rộng của sân khấu đều nhỏ hơn hai lần so với chế độ mặc định. Để chuyển sân khấu về chế độ này, các bạn vào mục **Chỉnh sửa** trên **Thanh danh mục**, chọn **Small stage layout**. Chọn lại một lần nữa để sân khấu chuyển về chế độ mặc định như ban đầu.

3. Chế độ toàn màn hình: Khu vực sân khấu được phóng to toàn màn hình. Để chuyển sân khấu về chế độ này, các bạn nhấp chuột vào  ở góc trên bên trái sân khấu.



Hình 1.3.5 . Chế độ chế độ toàn màn hình và chế độ ngầm định của sân khấu

Dù ở chế độ nào thì hệ tọa độ trên sân khấu vẫn không thay đổi, trục x từ -240 đến 240 , trục y từ -180 đến 180 .

Ở góc trên bên phải sân khấu có hai nút  và .

Khi bạn chọn nút lệnh  thì các khối lệnh được ghép với khối lệnh  khi nhấn vào trong nhóm Chọn sự kiện (Khu lập trình) sẽ được thực hiện. Biểu tượng này thường dùng để kích hoạt khối lệnh bắt đầu dự án.

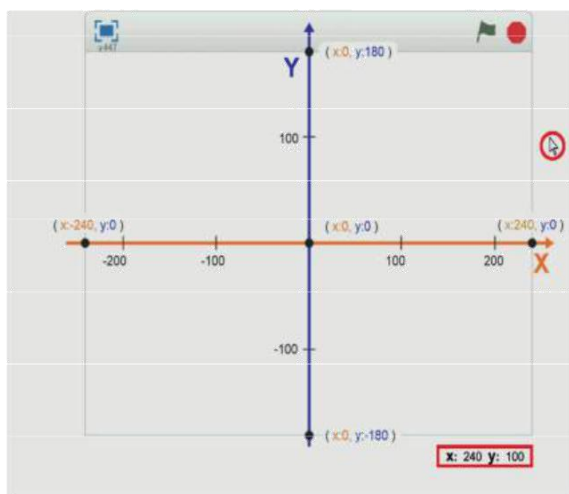
Khi chọn nút lệnh  thì tất cả hoạt động trên sân khấu sẽ dừng lại. Có thể ngừng chương trình vào thời điểm bất kì.

Khi chương trình đang chạy thì biểu tượng lá cờ xanh có dạng  (sáng lên), trong khi biểu tượng **Dừng** có dạng  (tối đi). Khi chương trình dừng thì ngược lại.



Hình 1.3.6. Biểu tượng Lá cờ xanh và Dừng

Phần cuối cùng trên sân khấu là vị trí tọa độ của con trỏ chuột. Vì giới hạn chiều ngang theo trục x của sân khấu là từ -240 đến 240 , giới hạn chiều dọc theo trục y là từ -180 đến 180 , nên khi di chuyển con trỏ chuột ra ngoài phạm vi sân khấu, tọa độ chỉ hiển thị giá trị lớn nhất (hoặc nhỏ nhất) của vị trí x và vị trí y .



Hình 1.3.7 . Toạ độ của con trỏ chuột



Tóm tắt lí thuyết và bài tập thực hành

Sân khấu - Stage

Sân khấu và hệ trục tọa độ

daylaptrinh.net

Số âm -3 -2 -1 0 1 2 3

Số âm là các số nhỏ hơn 0.
Ví dụ: -1; -2; -3;.....

Toạ độ

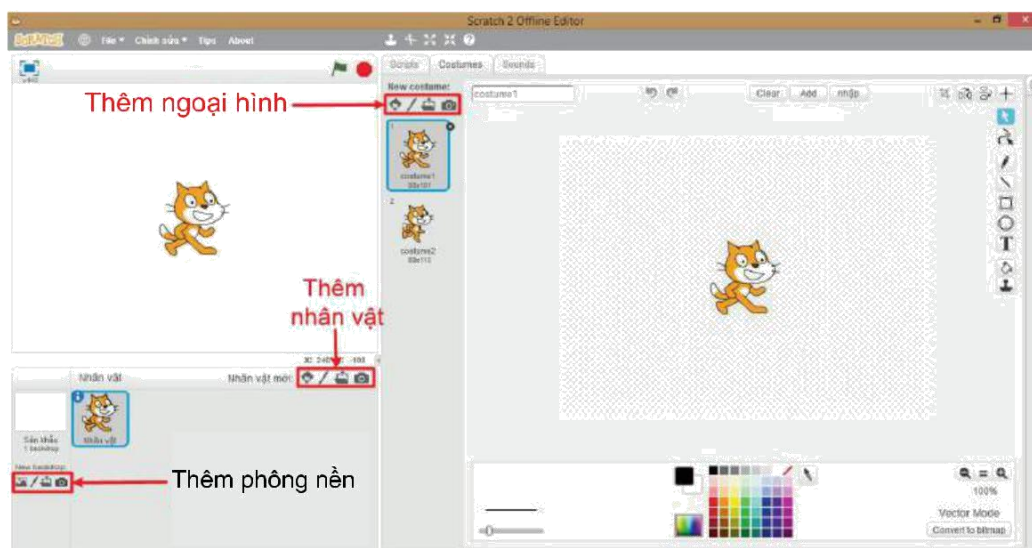
Toạ độ A (100;100)

Bài tập

Xác định toạ độ các điểm sau:

BÀI 4 LÀM VIỆC VỚI PHÒNG NỀN, NHÂN VẬT VÀ NGOẠI HÌNH NHÂN VẬT

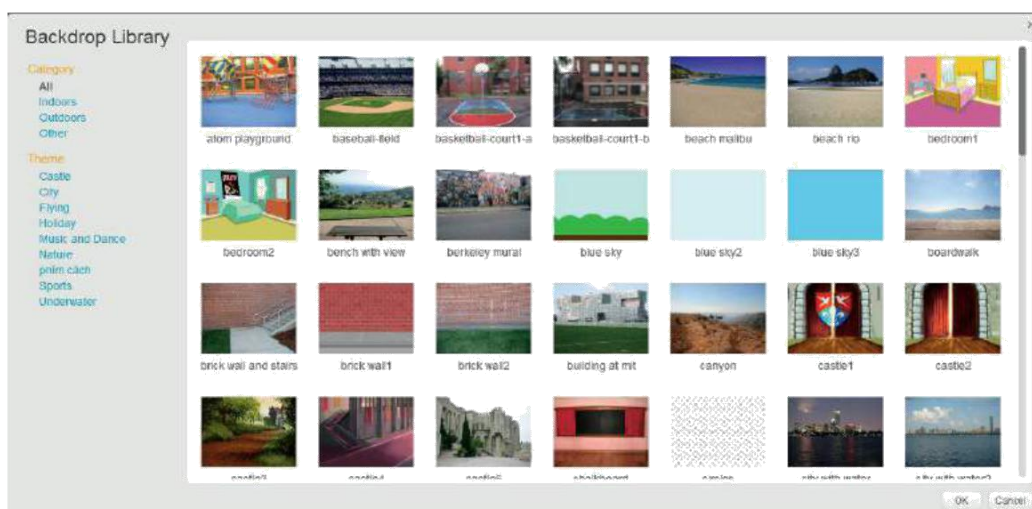
Việc thêm phòng nền, nhân vật và ngoại hình nhân vật đều có 4 cách:



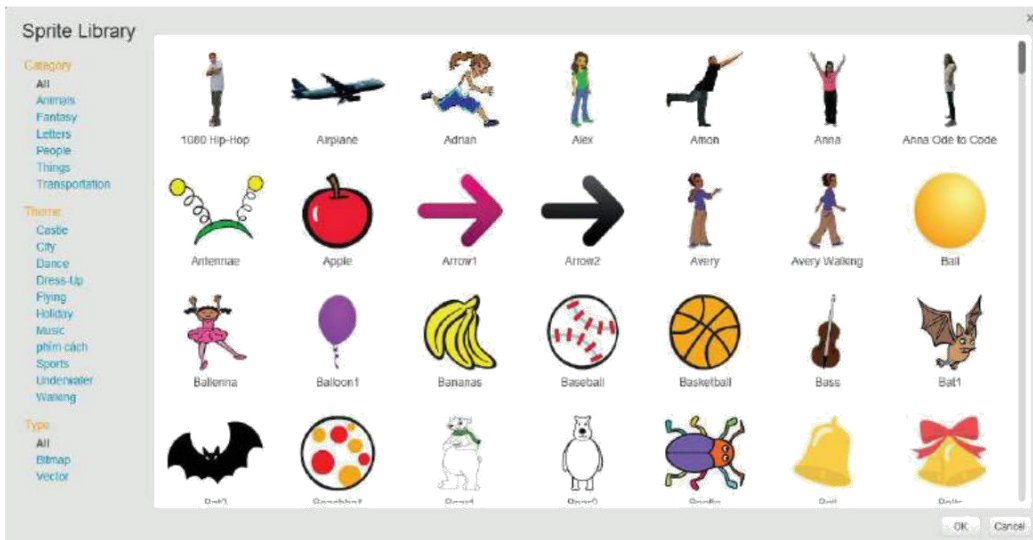
Hình 1. .1. Thêm hông nề , nhân vậ và ngoại hình nhân vật

4.1. Chọn đối tượng từ thư viện

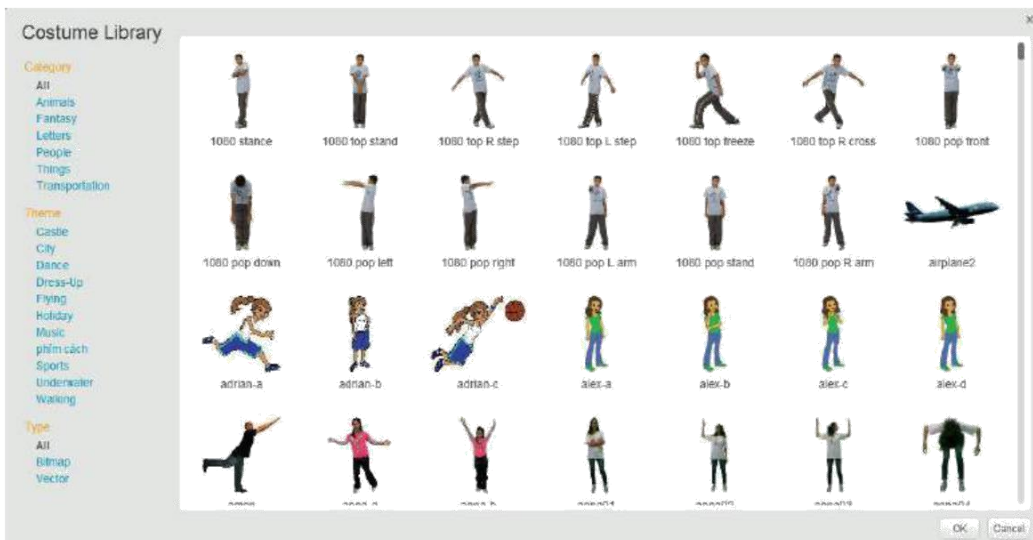
Biểu tượng của lệnh cho phép Chọn đối tượng từ thư viện khi muốn thay phòng nền là và khi muốn thay nhân vật là . Bạn nháy chuột vào một trong hai biểu tượng đó sẽ mở ra thư viện của hình nền, nhân vật và ngoại hình tương ứng, cùng với nhiều thể loại và chủ đề khác nhau.



Hình 1.4.2. Thư viện phòng nền



Hình 1.4.3. Thư viện nhân vật



Hình 1.4.4. Thư viện ngoại hình

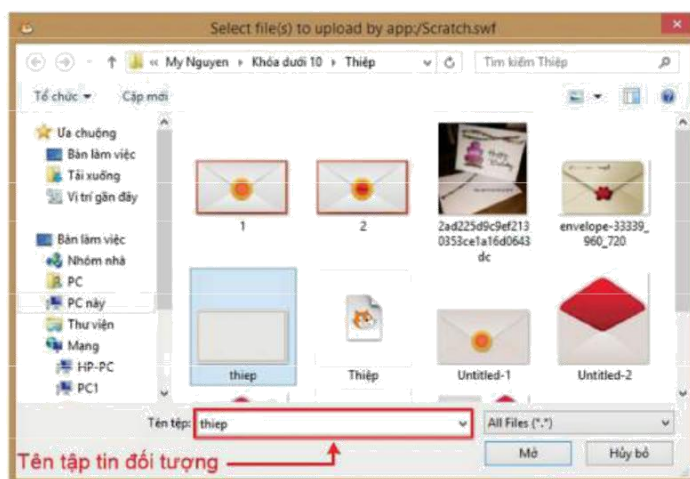
4.2. Vẽ mới

Khi chọn lệnh để vẽ mới , Khu thiết kế xuất hiện cùng với các công cụ để bạn tự vẽ, tự thiết kế đối tượng mới cho riêng mình. Chúng ta sẽ làm quen với Khu thiết kế ở bài học sau.

4.3. Tải đối tượng từ máy tính

Nếu bạn muốn thêm một đối tượng có sẵn từ máy tính, chọn lệnh  để tải từ máy tính. Một cửa sổ xuất hiện để các bạn chọn ảnh hoặc tập tin đối tượng muốn thêm vào chương trình.

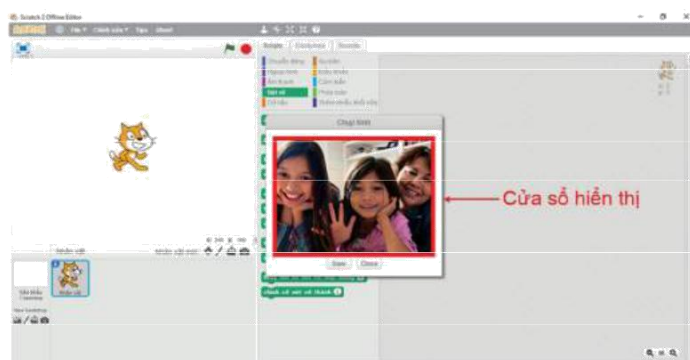




Hình 1.4.5. Cửa sổ thêm đối tượng từ máy tính

4.4. Chụp ảnh

Các bạn có thể chọn lệnh Chụp ảnh  để chụp hình những đối tượng trên màn hình và lưu làm đối tượng mới.



Hình 1.4.6. Chụp ảnh từ máy tính

Lưu ý: Trong trường hợp muốn xóa phông nền hoặc hình ảnh của đối tượng trong danh sách, chọn phông nền hay hình ảnh muốn xóa và nhấp chuột vào  hoặc nhấp nút phải chuột vào hình ảnh muốn bỏ và chọn **Xoá**.



Hình 1.4.7. Xóa đối tượng trong danh sách





Tóm tắt lý thuyết và bài tập thực hành

Phông nền - Backdrop

Thay đổi phông nền cho dự án

daylaptrinh.net

Thay đổi phông nền

Thư viện | Vẽ mới | Tải lên | Chụp hình

Xoá

Nhảy chuột

Bài tập

- Thay đổi phông nền theo Hình 1 và Hình 2.
- Tự thay đổi phông nền theo ý thích của mình.

Hình 1

Hình 2

Nhân vật - Sprite

Thêm nhân vật cho dự án

daylaptrinh.net

Thêm từ thư viện

Cách thêm khác

Vẽ mới nhân vật

Chụp hình

Tải ảnh

Bài tập

- Thêm nhân vật, thay đổi phông nền để được hình bên:
- Tự thay đổi phông nền, thêm nhân vật theo ý thích.

Trang phục - Costume

Thay đổi, xoá ngoại hình cho nhân vật, phông nền sân khấu

daylaptrinh.net

Nhân vật

Thư viện | Vẽ mới | Tải lên | Chụp hình

Phông nền

Thư viện | Vẽ mới | Tải lên | Chụp hình

Xoá

tạo bản sao

xoá

save to local file

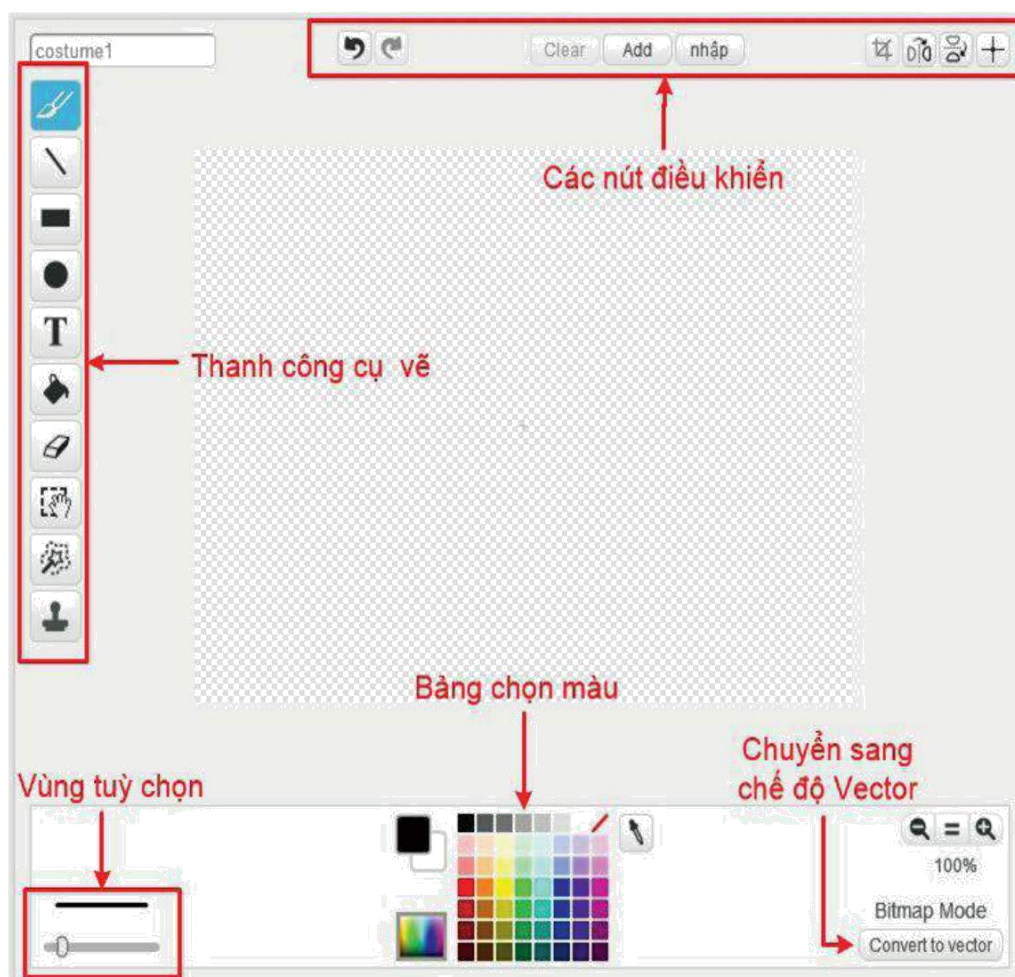


BÀI 5

SỬ DỤNG CÔNG CỤ VẼ

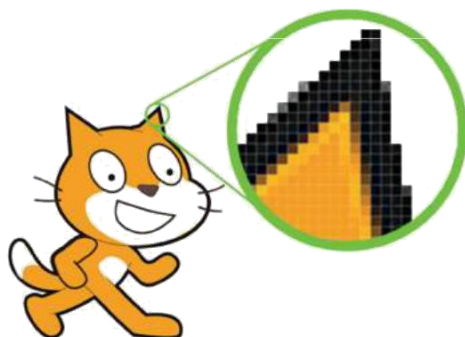
Phông nền hoặc trang phục sẽ được vẽ mới và chỉnh sửa ở Khu thiết kế, nơi có các công cụ vẽ để các bạn vẽ hay chỉnh sửa theo ý muốn. Khu thiết kế có hai chế độ vẽ: Chế độ Bitmap và chế độ Vector.

5.1. Chế độ Bitmap



Hình 1.5.1. Chế độ Bitmap

Ở chế độ Bitmap, các hình được tạo nên từ những chấm màu gọi là điểm ảnh (tiếng Anh: pixel).



Hình 1.5.2. Các điểm ảnh ở chế độ Bitmap

Khi phóng to ảnh, chất lượng ảnh sẽ bị giảm, ảnh sẽ nhoè đi. Hiện tượng này được gọi là vỡ ảnh.



Hình 1.5.3. Hiện tượng vỡ ảnh

Các bạn có thể thấy ở chế độ này, khu thiết kế gồm các thành phần:

1. Các nút điều khiển

Các nút điều khiển ở bên trên khu thiết kế với các chức năng:

	Hủy bỏ thao tác vừa thực hiện.
	Thực hiện lại thao tác vừa mới bị hủy.
	Xoá tất cả đối tượng được vẽ trên khu thiết kế.
	Thêm ảnh có sẵn trong thư viện.
	Thêm ảnh có sẵn được lưu trên máy tính.
	Lật ảnh sang trái/phải.
	Lật ảnh lên trên/xuống dưới.
	Điều chỉnh tâm của nhân vật.

2. Bảng chọn màu

Để có thể chọn được màu cho bút vẽ hay các công cụ vẽ hình vẽ, đoạn thẳng..., các bạn có thể sử dụng bảng màu có sẵn hoặc dùng công cụ lấy màu.

Bảng màu có sẵn được chia ra làm hai loại: bảng màu cơ bản và bảng màu nâng cao. Bảng màu cơ bản chỉ có 56 màu, trong khi bảng màu nâng cao có hàng nghìn màu và có thêm thanh chỉnh độ sáng.





Hình 1.5.4. Hai loại bảng màu có sẵn

Trong trường hợp muốn chọn một màu trên hình vẽ trong Khu thiết kế, các bạn nhấp chuột vào công cụ  (bên phải bảng màu), di chuyển con trỏ chuột và nhấp vào vùng có màu muốn lấy.



Hình 1.5.5. Công cụ lấy màu

3. Thanh công cụ vẽ

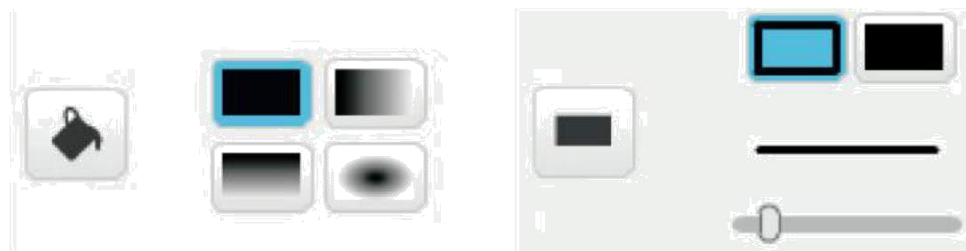
Thanh công cụ vẽ gồm các công cụ:

-  Vẽ một hình bất kì.
-  Vẽ đoạn thẳng theo ý muốn.
-  Vẽ hình chữ nhật. Khi muốn vẽ hình vuông nhấn giữ phím **Shift** trong quá trình thao tác chuột.
-  Vẽ hình elip. Khi muốn vẽ hình tròn, nhấn giữ phím **Shift** trong khi thao tác chuột.
-  Thêm văn bản (chữ) vào hình vẽ.
-  Tô màu một vùng khép kín nhất định, như hình vuông, hình chữ nhật, hình tròn,...
-  Tẩy, xoá hình.
-  Chọn một vùng tùy thích.
-  Xoá phần nền ngoài phần chọn của chuột.
-  Sao chép vùng được chọn.

Các bạn chú ý rằng nếu muốn thay đổi kích thước nét vẽ hoặc màu vẽ cho hình, các bạn cần phải thay đổi trước khi vẽ. Ở chế độ Bitmap thì hình sau khi vẽ sẽ không thể thay đổi màu sắc hay kích thước nét vẽ. Việc thay đổi vị trí cũng rất giới hạn, các bạn chỉ có thể thay đổi ngay sau khi vẽ, nếu chuyển sang vẽ hình mới thì sẽ không thể di chuyển hình cũ.

4. Vùng tùy chọn

Vùng tùy chọn ở phía dưới bên trái khu thiết kế. Với mỗi công cụ vẽ khác nhau, vùng này lại hiển thị chức năng tương ứng, ví dụ như thay đổi kích thước nét vẽ, kiểu vẽ, thay đổi phông chữ,...



Hình 1.5.6. Các công cụ với vùng tùy chọn tương ứng

5.2. Chế độ Vector



Hình 1.5.7. Chế độ Vector

Ở chế độ Vector, khi phóng to ảnh không bị nhoè hay vỡ ảnh mà vẫn rõ ràng sắc nét.



Hình 1.5.8. Phóng to ảnh trong chế độ Vector

Các nút điều khiển, bảng màu và vùng tùy chọn trong chế độ Vector đều giống với chế độ Bitmap. Chỉ có thanh công cụ vẽ là khác cả về vị trí và thành phần.

-  Chọn hình vẽ.
-  Chỉnh hình, thay đổi hình dáng nét vẽ.
-  Vẽ mọi hình bất kỳ
-  Vẽ đoạn thẳng theo ý muốn.
-  Vẽ hình vuông hoặc hình chữ nhật.
-  Vẽ hình elip hoặc hình tròn.
-  Thêm văn bản.
-  Tô màu một vùng khép kín nhất định.
-  Tạo bản sao của hình vẽ.
-  Đặt hình ra trước một lớp.
-  Đặt hình ra sau một lớp.
-  Tách nhóm.
-  Gộp các hình thành nhóm.

Ở chế độ Vector thì các hình vẽ có thể được thay đổi vị trí và kích thước tùy ý sau khi vẽ. Dòng văn bản ở chế độ này cũng có thể được thay đổi nội dung và màu sắc bất kỳ lúc nào.



Tóm tắt lí thuyết và bài tập thực hành

Ngoại hình - Costumers

Vẽ ở chế độ Bitmap

daylaptrinh.net

Vẽ tự do Vẽ đường thẳng Vẽ hình chữ nhật Vẽ hình elip

Viết chữ Tô màu Tẩy xóa Chọn Xóa nền

Bài tập

Dùng công cụ ở chế độ Bitmap tạo sân khấu cho riêng mình.
(Tham khảo hình bên)

Ngoại hình - Costumers

Vẽ ở chế độ Vector

daylaptrinh.net

Chọn Chính hình Vẽ tự do Vẽ đường thẳng

Vẽ hình chữ nhật Vẽ hình elip Viết chữ Tô màu

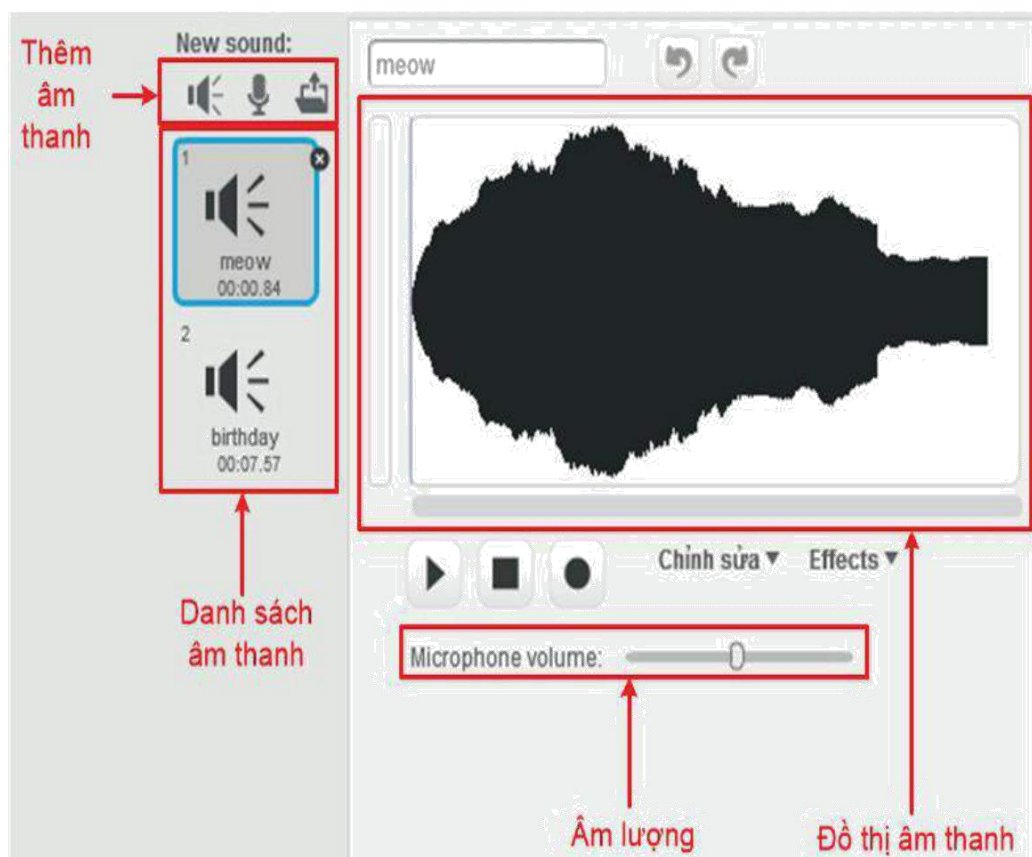
Bài tập

- Dùng các công cụ ở chế độ vector để vẽ chiếc cốc.
- Dùng các công cụ ở chế độ vector tạo ra nhân vật hoặc tạo ra phòng nền.

BÀI 6

KHU ÂM THANH

Chương trình của các bạn sẽ trở nên thú vị hơn rất nhiều nếu có thêm âm thanh. Âm thanh của nhân vật được quản lý trong trang Âm thanh (Sounds).



Hình 1.6.1. Khu âm thanh

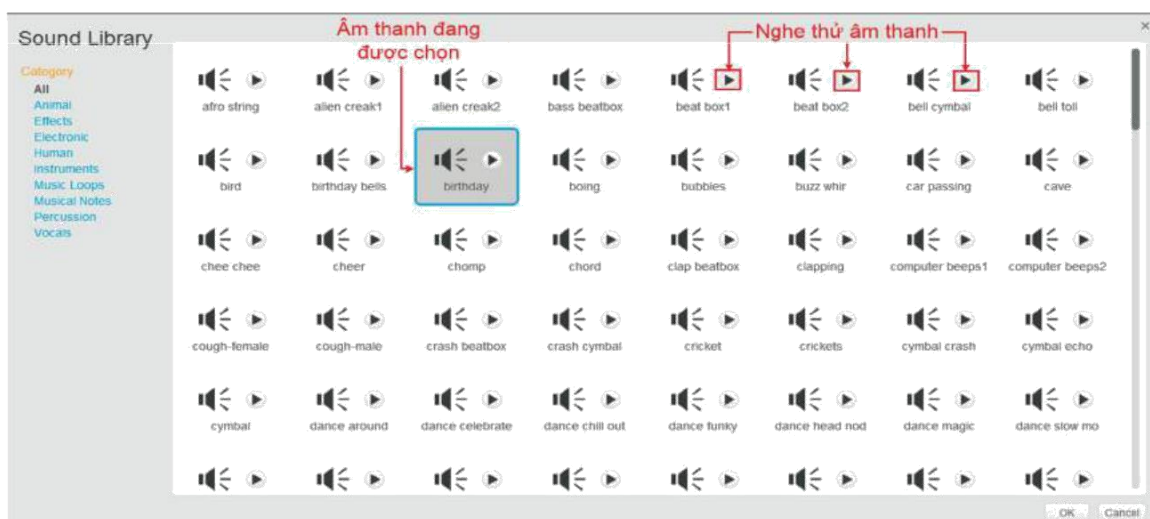
6.1. Thêm và xoá âm thanh

Để thêm âm thanh cho mỗi đối tượng trong chương trình, chúng ta có ba cách:

1. Chọn từ thư viện

Khi nháy chuột chọn , thư viện âm thanh sẽ xuất hiện. Âm thanh trong thư viện được chia thành nhiều nhóm khác nhau để các bạn có thể tìm kiếm và lựa chọn nhanh chóng.

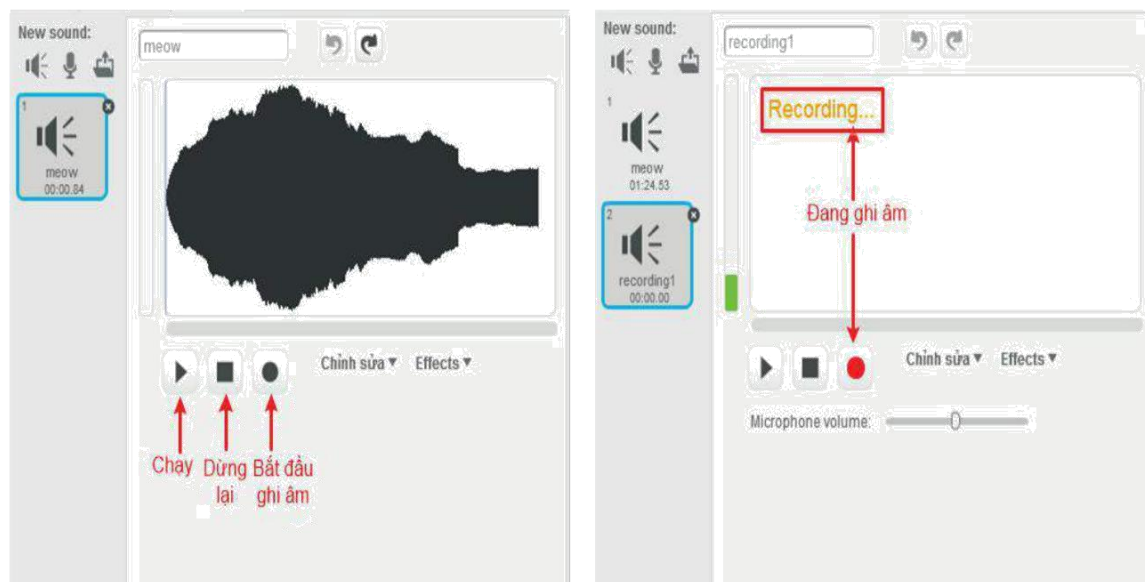




Hình 1.6.2. Thư viện âm thanh

2. Ghi âm

Các bạn có thể tự ghi âm bằng cách chọn lệnh . Lúc này một tệp âm thanh mới sẽ được tạo ra bằng cách thu âm thanh qua microphone của máy tính. Các bạn nhấp chuột vào nút để tiến hành ghi âm.

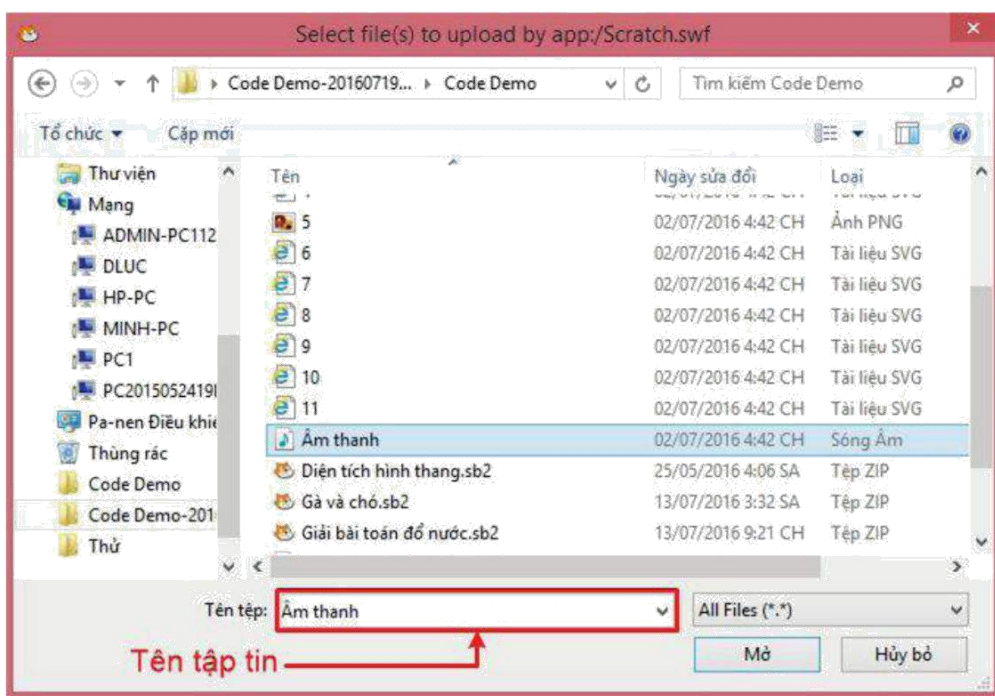


Hình 1.6.3. Ghi âm

3. Tải từ máy tính

Nếu các bạn muốn thêm âm thanh hay nhạc có sẵn ở trong máy tính, chọn lệnh và một cửa sổ hiện ra cho phép chọn tệp tin muốn thêm.





Hình 1.6.4. Cửa sổ thêm âm thanh từ máy tính

Trong trường hợp các bạn muốn bỏ âm thanh trong danh sách âm thanh, chọn âm thanh muốn bỏ và hãy chuột vào nút hoặc nhấn nút phải chuột vào âm thanh muốn bỏ và chọn **Xoá**.



Hình 1.6.5. Xoá âm thanh trong danh sách

6.2. Chỉnh sửa và thêm hiệu ứng cho âm thanh

Khu âm thanh hỗ trợ rất nhiều thao tác chỉnh sửa cũng như thêm hiệu ứng cho âm thanh. Trong khu có danh mục **Chỉnh sửa** và **Effects** (Tạo hiệu ứng) để chúng ta làm việc với các đối tượng âm thanh, cụ thể như sau:





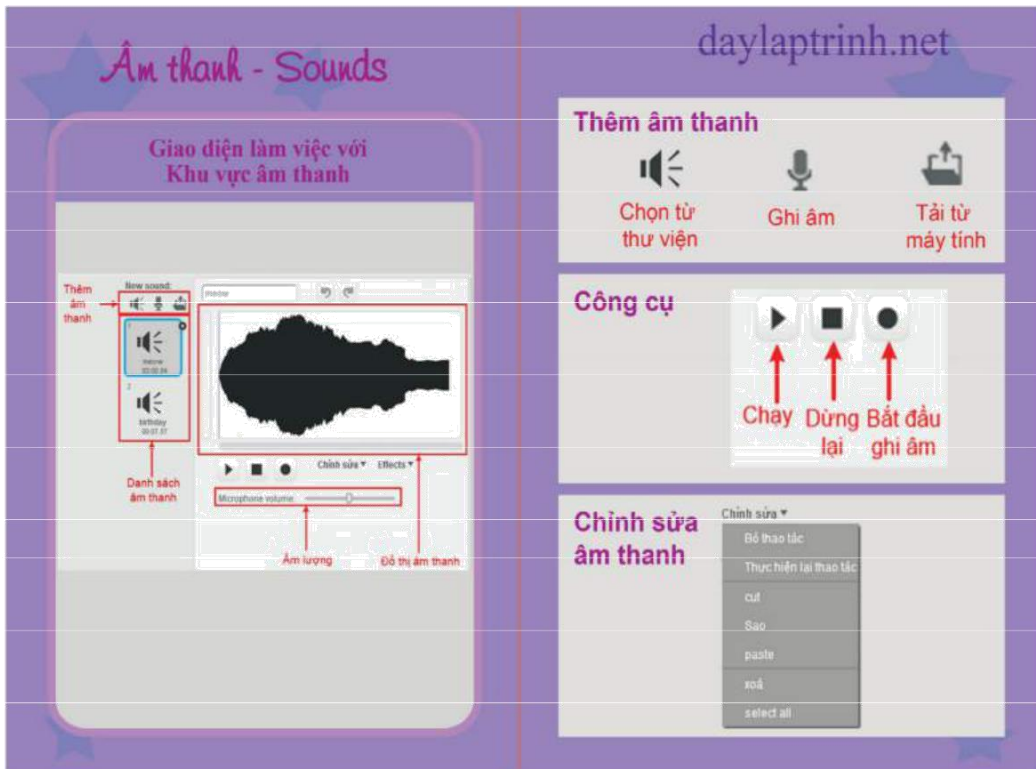
Hình 1.6.6. Chỉnh sửa âm thanh



Hình 1.6.7. Thêm hiệu ứng cho âm thanh



Tóm tắt lí thuyết và bài tập thực hành



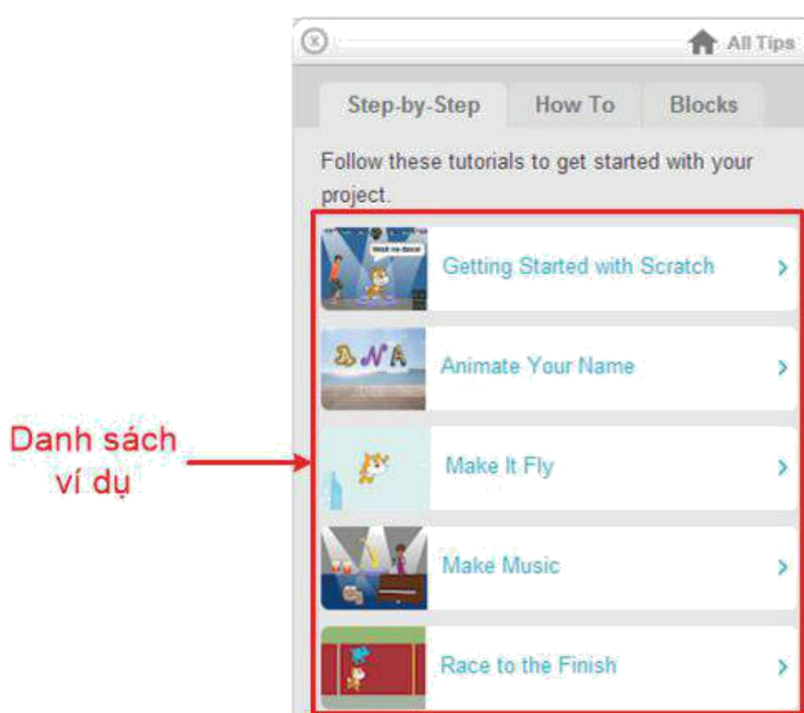
BÀI 7

CỬA SỔ TRỢ GIÚP

Cửa sổ trợ giúp chứa rất nhiều hướng dẫn và ví dụ mẫu. Đây là khu vực rất hữu ích, giúp các bạn hiểu ý nghĩa những khối lệnh, hướng dẫn một số hiệu ứng, chuyển động, trò chơi,... Tuy nhiên, ở phiên bản v445 thì khu vực này chưa được Việt hoá, nên có thể gây một số khó khăn cho các bạn.

Cửa sổ trợ giúp được chia làm ba trang:

7.1. Step-by-Step

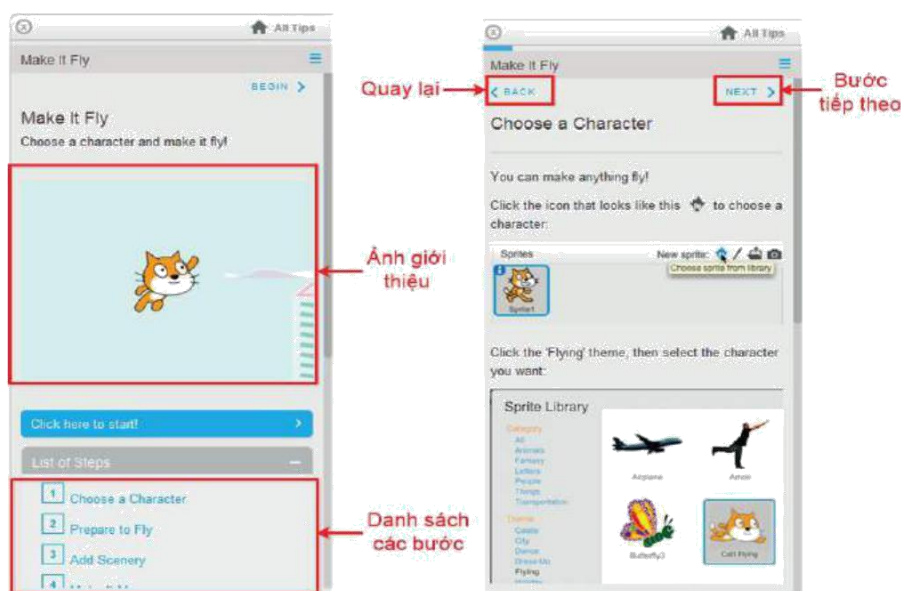


Hình 1.7.1. Trang **Step-by-Step** trong Cửa sổ trợ giúp

Trang này sẽ hướng dẫn cho các bạn từng bước để tạo ra một ví dụ mẫu.

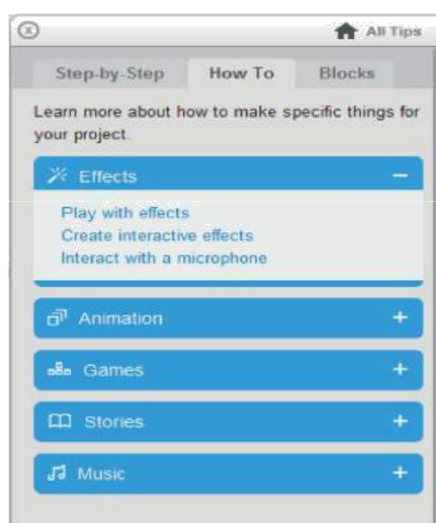
Chọn một ví dụ bất kì, một ảnh động giới thiệu về ví dụ mẫu và danh sách các bước. Khi mở danh sách và chọn một bước bất kì, clip hướng dẫn chi tiết của bước đó sẽ hiện ra.





Hình 1.7.2. Giới thiệu về ví dụ mẫu và hướng dẫn chi tiết mỗi bước

7.2. How To



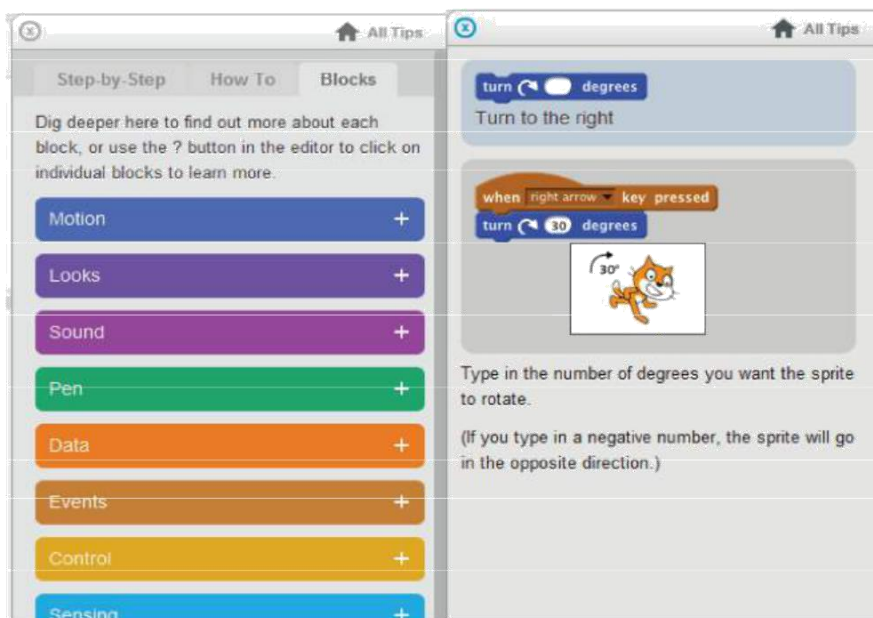
Hình 1.7.3. Trang **How To** trong Cửa sổ trợ giúp

Trang này chứa những hướng dẫn các kĩ thuật thiết yếu cho mỗi chương trình, được chia theo năm chủ đề:

- **Effects:** Thêm hiệu ứng cho nhân vật.
- **Animation:** Tạo các hành động cho nhân vật như nói, di chuyển,...
- **Games:** Những hướng dẫn liên quan đến thiết kế trò chơi, ví dụ như điều khiển nhân vật bằng bàn phím, tương tác giữa các nhân vật,...
- **Stories:** Những hướng dẫn trong việc thiết kế một câu chuyện.
- **Music:** Hỗ trợ cách thêm âm thanh, chỉnh sửa âm thanh, chơi một số nhạc cụ,...



7.3. Blocks

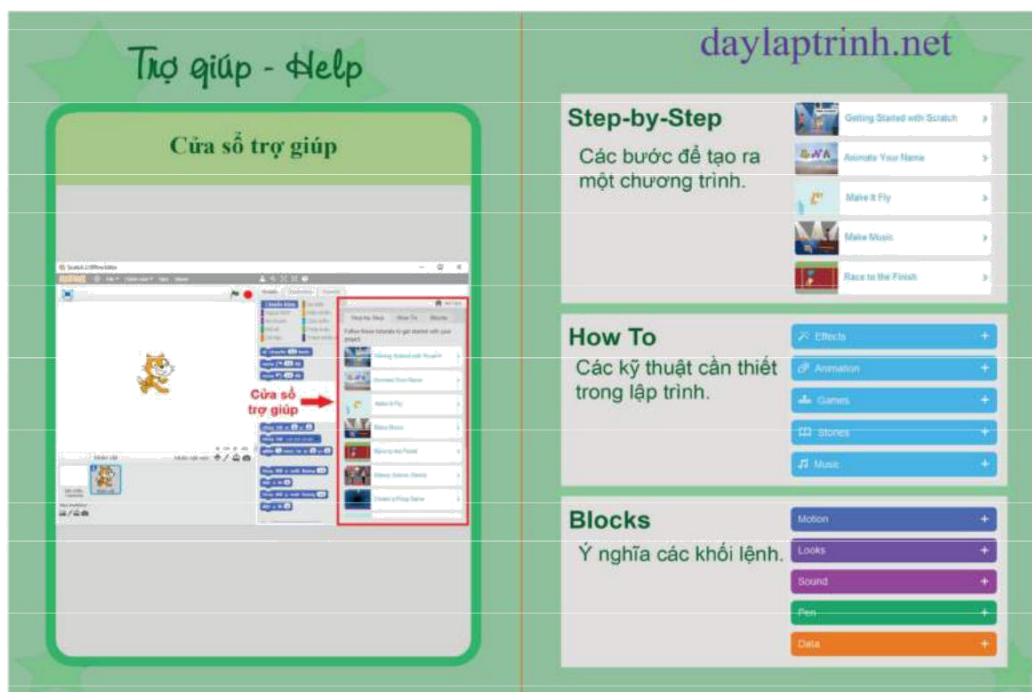


Hình 1.7.4. Trang **Blocks**

Trang này giải thích ý nghĩa của tất cả các khối lệnh và được chia nhóm tương tự như bên bảng khối lệnh. Khi chọn khối lệnh nào thì các bạn sẽ thấy hướng dẫn và ví dụ mẫu thực hiện khối lệnh đó.



Tóm tắt lí thuyết và bài tập thực hành



BÀI 8

KHU LẬP TRÌNH

Khu lập trình là nơi chứa các khối lệnh và ghép chúng lại với nhau tạo thành một chương trình theo ý muốn của lập trình viên. Trong bất kì dự án nào, chúng ta đều phải thực hiện các thao tác trên khu vực này. Khu lập trình nằm ở trang **Scripts**, chứa Bảng danh mục các khối lệnh.

8.1. Bảng danh mục các khối lệnh



Hình 1.8.1. Bảng danh mục các khối lệnh

Bảng danh mục các khối lệnh gồm các nhóm khối lệnh (gọi tắt là nhóm lệnh) và danh sách các khối lệnh tương ứng với từng nhóm. Scratch phiên bản 2.0 có hơn 100 khối lệnh được chia thành 10 nhóm. Các khối lệnh cùng một nhóm sẽ có màu sắc giống nhau.

Có bốn loại khối lệnh chính trong Scratch:

- 1. Khối lệnh thực hiện:** Có hình dạng  với khớp nối ở cả phía trên và phía dưới, dùng để thực hiện một công việc nhất định.
- 2. Khối lệnh giá trị:** Có hình dạng  hoặc , không có khớp nối nên không thể ghép vào phía trước hoặc phía sau khối lệnh khác được. Chúng lưu trữ một giá trị nhất định, có thể là số, xâu kí tự hoặc giá trị Đúng/Sai.
- 3. Khối lệnh kích hoạt:** Có hình dạng  và chỉ có khớp nối phía dưới. Chúng đợi một sự kiện nào đó (như nhấn phím trên bàn phím hoặc khi nháy chuột vào ) và sẽ thực hiện các khối lệnh bên dưới khi sự kiện đó xảy ra.
- 4. Khối lệnh điều khiển:** Có hình dạng . Chúng không những có khớp nối ở phía trên và phía dưới mà còn có khớp nối bên trong, tức là có thể bao lấy

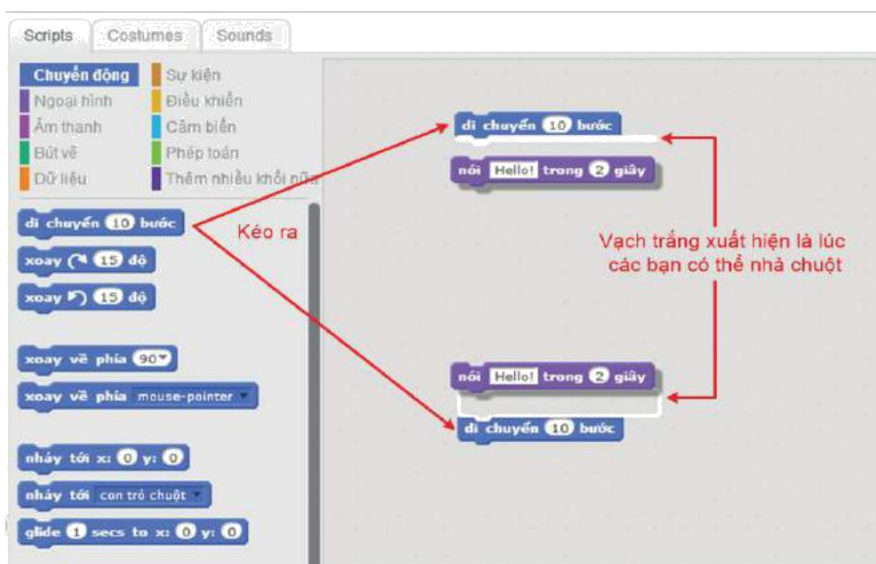
một hoặc một đoạn khối lệnh khác bên trong. Thông thường khi chạy một đoạn khối lệnh thì các khối lệnh sẽ được thực hiện theo thứ tự lần lượt từ trên xuống dưới, Khối lệnh điều khiển có thể làm thay đổi thứ tự thực hiện này.

Các bạn có thể nhấp chuột trực tiếp vào khối lệnh để chạy khối lệnh đó. Khối lệnh sẽ thực hiện công việc sau khi nhấp chuột 1 giây. Khối lệnh giá trị sẽ hiển thị cho chúng ta giá trị đang lưu trữ.



Hình 1.8.2. Thực hiện các khối lệnh bằng cách nhấp chuột

Bên cạnh bảng danh mục các khối lệnh là nơi mà các bạn kéo thả các khối lệnh để ghép chúng lại với nhau. Khi ghép các khối lệnh, các bạn cần phải nhấn giữ và kéo chuột cho đến khi giữa hai khối lệnh xuất hiện vạch sáng màu trắng thì mới thả chuột. Lúc này hai khối lệnh sẽ được ghép lại với nhau.



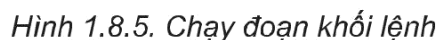
Hình 1.8.3. Ghép các khối lệnh thực hiện

Ghép các khối lệnh kích hoạt, khối lệnh sự kiện, khối lệnh giá trị cũng tương tự như vậy.



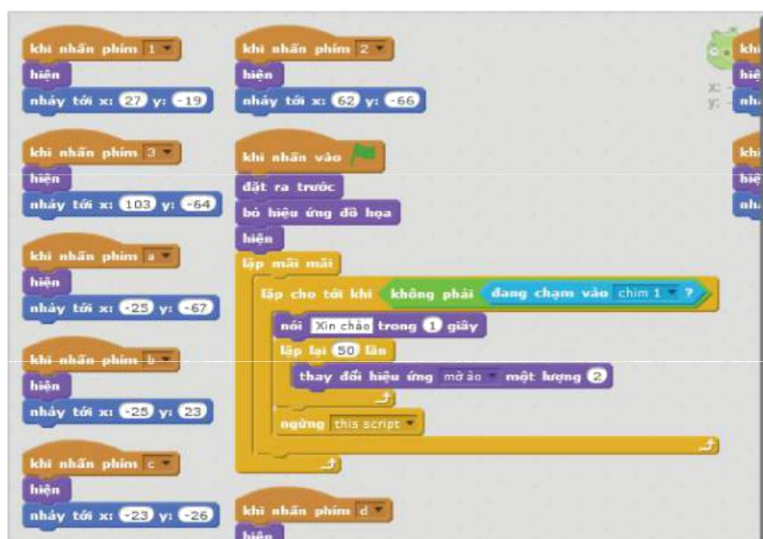
Hình 1.8.4. Ghép các khối lệnh khác



Tải miễn phí tại: Webtietkiem.com

Vì các bạn có thể kéo thả các khối lệnh và đặt ở bất cứ đâu trong khu lập trình nên nơi này rất dễ ộn xộn và ối mắt. ể dọn dẹp , sắp xếp lại các khối lệnh cho ngăn nắp hơn, các bạn nháy nút phải chuột vào một khoảng trống bất kì trong khu lập trình và chọn **Clean up.**





b) Sau khi dọn dẹp Khu lập trình

Hình 1.8.6.

Khi nhấp nút phải chuột vào một khối lệnh bất kì ở trong khu lập trình, các bạn cũng có một số lựa chọn, chẳng hạn chọn **add comment** để viết thêm nhận xét mới hoặc chú thích cho lệnh.

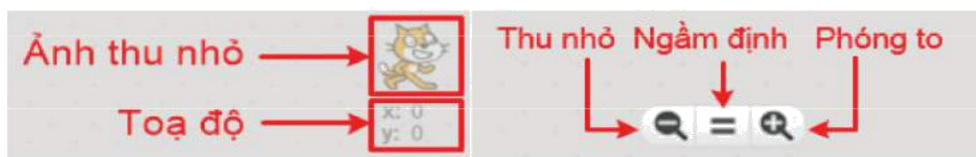


Hình 1.8.7. Các lựa chọn khi nhấp nút phải chuột vào khối lệnh

- **Tạo bản sao:** tạo ra một bản sao của khối lệnh và đoạn khối lệnh gắn theo sau. Bản sao này sẽ dính vào con trỏ chuột để các bạn di chuyển cho đến khi nhấp chuột tại bất kì vị trí trên khu lập trình, thậm chí là sang nhân vật hay phong nền khác.
- **Xóa:** xóa khối lệnh và đoạn khối lệnh theo sau.
- **Add comment:** thêm chú thích cho khối lệnh.
- **Help:** mở phần trợ giúp về khối lệnh.

Ở góc trên bên phải khu lập trình là ảnh thu nhỏ và tọa độ của nhân vật, giúp các bạn theo dõi dễ dàng hơn, còn ở góc dưới bên phải là công cụ để phóng to, thu nhỏ các khối lệnh.





Hình 1.8.8. Công cụ Khu lập trình



Tóm tắt lí thuyết và bài tập thực hành

Khu lập trình

Không gian làm việc

daylaptrinh.net

Ghép các khối lệnh

Dọn dẹp và thêm chú thích

clean up → Dọn dẹp

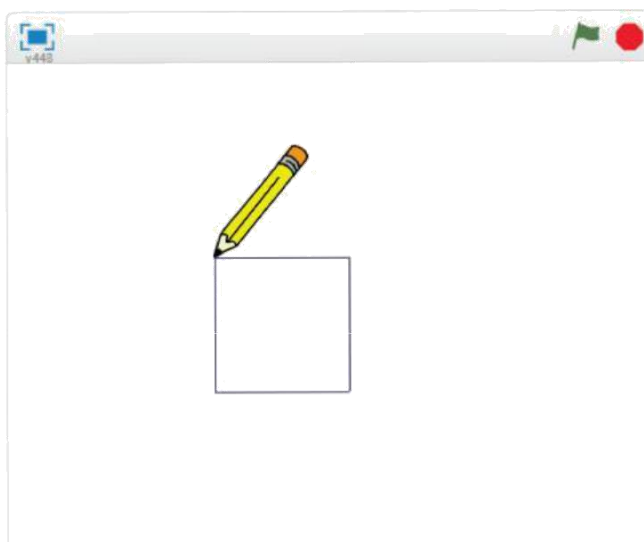
add comment → Thêm chú thích

Công cụ

BÀI 9

VÍ DỤ CƠ BẢN

Sau đây sẽ là một ví dụ vẽ hình vuông đơn giản. Chúng ta cần điều khiển nhân vật chiếc bút để vẽ được một hình vuông như trong *Hình 1.9.1*.



Hình 1.9.1. Ví dụ vẽ hình vuông

9.1. Thêm mới nhân vật

Để thay đổi nhân vật ngầm định là chú mèo thành nhân vật chiếc bút, chúng ta cần xoá chú mèo đi và thêm chiếc bút từ thư viện.

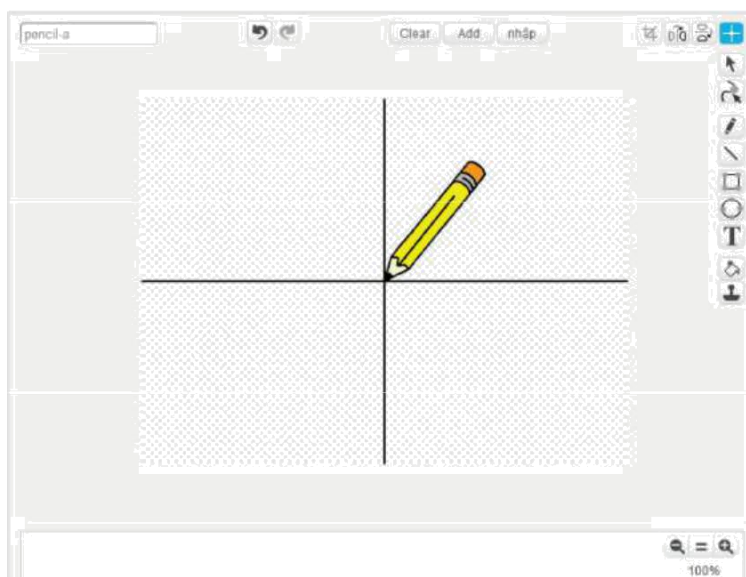


Hình 1.9.2. Thay đổi nhân vật

9.2. Thay đổi tâm nhân vật

Khi các bạn vẽ, nét bút sẽ xuất hiện từ tâm của nhân vật. Ngầm định tâm của nhân vật sẽ ở giữa, vì vậy ta cần thay đổi lại vị trí tâm của chiếc bút lên đầu bút để nét vẽ có thể xuất hiện trên đầu bút bằng công cụ  bên khu ngoại hình (Costume).



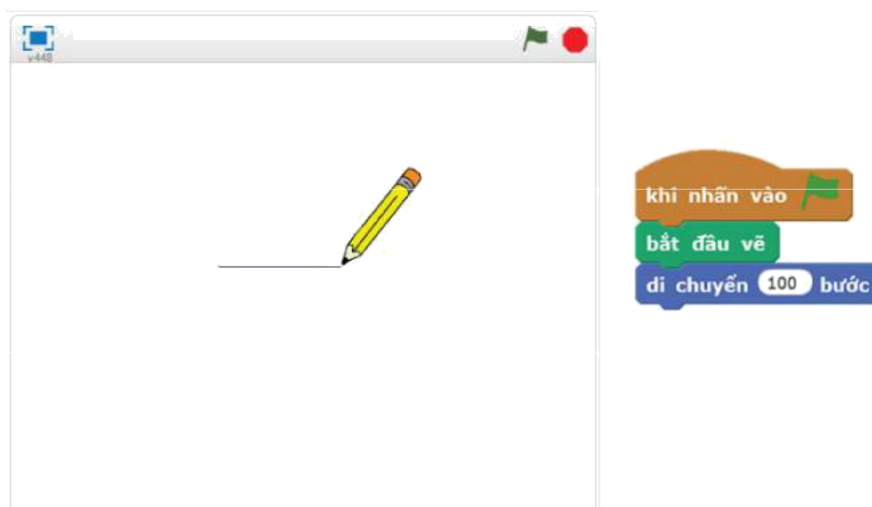


Hình 1.9.3. Thay đổi tâm của nhân vật

9.3. Vẽ cạnh đầu tiên

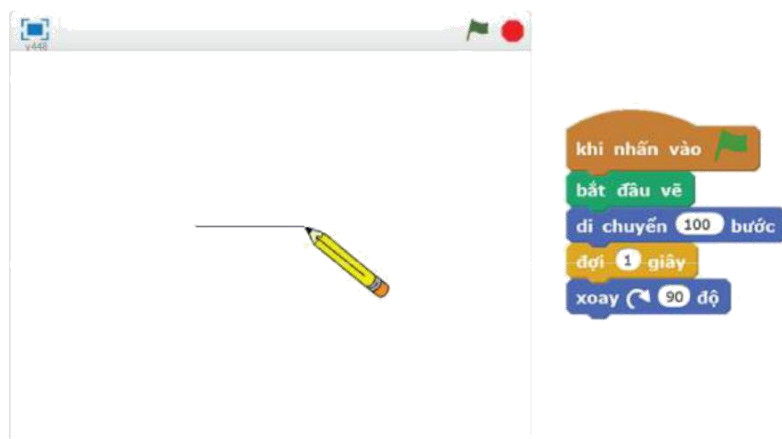
Đầu tiên, các bạn cần khối lệnh **khi nhấn vào** và **bắt đầu vẽ**. Sau khối lệnh này, việc di chuyển của nhân vật sẽ kẻ theo nét vẽ xuất hiện từ tâm của nhân vật. Tiếp theo các bạn sử dụng khối lệnh **di chuyển 10 bước** và thay đổi thành 100 bước để vẽ cạnh đầu tiên bằng cách nháy chuột vào ô **10** sau đó sửa thành 100.

Lưu ý: Chọn khối lệnh theo màu sắc của nhóm lệnh.



Hình 1.9.4. Vẽ cạnh hình vuông đầu tiên

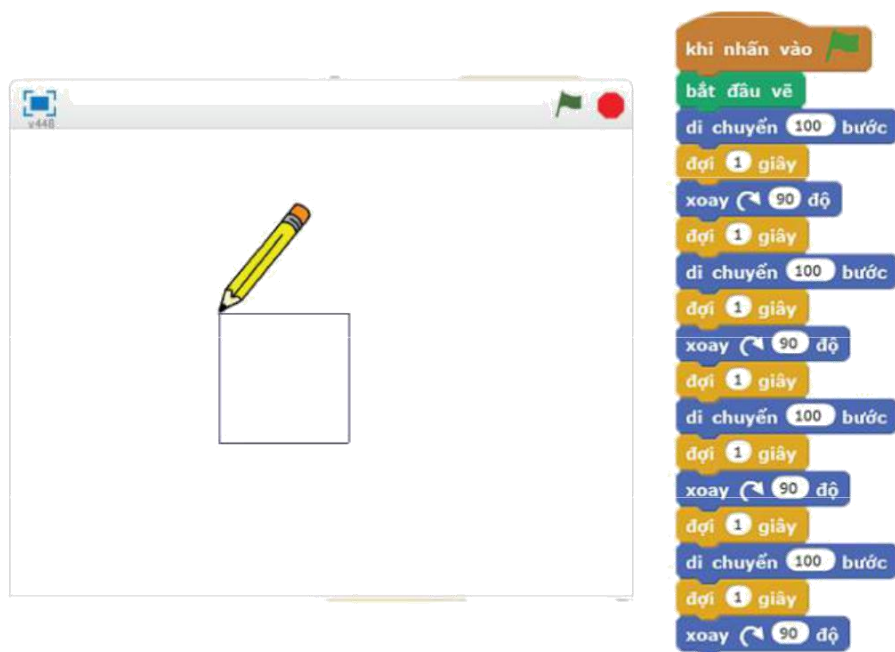
Sau khi vẽ cạnh đầu tiên, các bạn cần đổi hướng chiếc bút xoay xuống dưới một góc vuông bằng 90° để vẽ cạnh tiếp theo bằng khối lệnh **xoay** 90° độ. Tuy nhiên, để quan sát chiếc bút di chuyển và xoay dễ dàng hơn, các bạn thêm khối lệnh **đợi 1 giây** vào giữa hai khối lệnh.



Hình 1.9.5. Khối lệnh xoay và đợi

9.4. Hoàn thành hình vuông

Thực hiện việc di chuyển và xoay để vẽ ba cạnh còn lại giống như cạnh đầu tiên.



Hình 1.9.6. Hoàn thành hình vuông



Tóm tắt lí thuyết và bài tập thực hành

Ví dụ cơ bản

Vẽ hình vuông

daylaptrinh.net

Tâm nhân vật

Đoạn khối lệnh

Bài tập

Thay đổi kích thước hình vuông



CHƯƠNG

II

Thiệp sinh nhật

Trong chương này, các bạn sẽ được hướng dẫn tạo ra một tấm thiệp sinh nhật đầy màu sắc và thật sinh động. Chúng ta sẽ tìm hiểu cách để thiết kế nhân vật, di chuyển nhân vật, thay đổi màu sắc, thêm âm thanh cho sinh động,... qua đó bước đầu hiểu được một số kiến thức lập trình cơ bản như thực hiện khối lệnh tuần tự, thông báo, sự kiện, cấu trúc vòng lặp,... Sau khi kết thúc chương, các bạn sẽ có thể tự tạo cho mình một tấm thiệp sinh nhật với các nhân vật, hiệu ứng, âm thanh tùy ý.

Tổng quan chương trình



Mọi chương trình đều được tạo nên bởi các **nhân vật** (hay còn gọi là **đối tượng**). Lập trình viên sẽ lập trình điều khiển các nhân vật đó thông qua các khối lệnh. Ví dụ làm quả bóng bay, chú chuột di chuyển, dòng chữ đổi màu, chiếc loa phát nhạc,...



Chương trình đầu tiên của chúng ta là tấm thiệp sinh nhật trong đó có các đối tượng:

1. Chong chóng

Chong chóng được lập trình quay liên tục cho tới khi dừng chương trình.



2. Que

Que có vị trí cố định, là đối tượng gắn với chong chóng, quy định vị trí của chong chóng.



3. Bóng bay

Khi chạy chương trình, Bóng bay xuất hiện tại vị trí ngẫu nhiên và bay từ dưới lên trên.



4. Chữ Happy



Chữ Happy có vị trí cố định trên sân khấu.

5. Chữ Bi thday

Chữ Birthday xuất hiện dưới chữ Happy và liên tục thay đổi màu.

Birthday

6. Nến

Nến được lập trình cháy liên tục cho đến khi dừng chương trình.



7. Mũ

Mũ sẽ di chuyển lên và xuống nhịp nhàng bên trên chữ P thứ 2 của chữ Happy.



8. Pháo

Pháo cũng di chuyển lên và xuống giống Mũ nhưng ngược với Mũ, khi Mũ di chuyển lên thì Pháo di chuyển xuống và ngược lại.



9. Ngôi sao

Ngôi sao sẽ tạo bản sao xung quanh phần trên của thiệp. Tất cả các bản sao này đều có hiệu ứng đổi màu và nhấp nháy.



BÀI 1

TẠO HÌNH NỀN VÀ VẼ CHONG CHÓNG

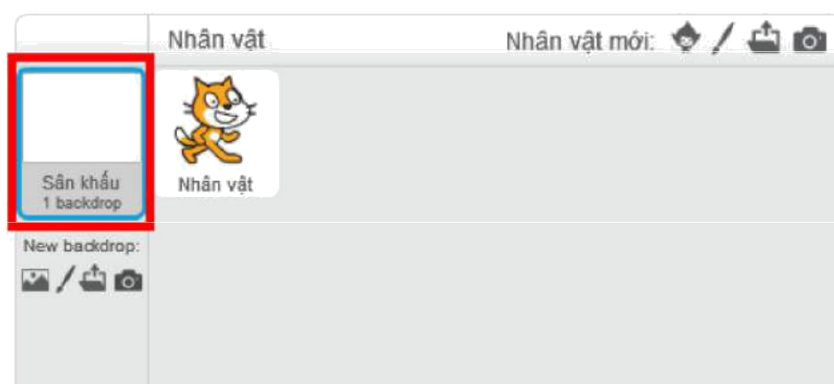
Mở phần mềm Scratch 2.0, tạo và lưu ứng dụng có tên “Thiệp”. Trong ứng dụng, trước tiên các bạn thiết kế hình nền cho thiệp.

1.1. Hình nền



Hình 2.1.1. Hình nền của thiệp

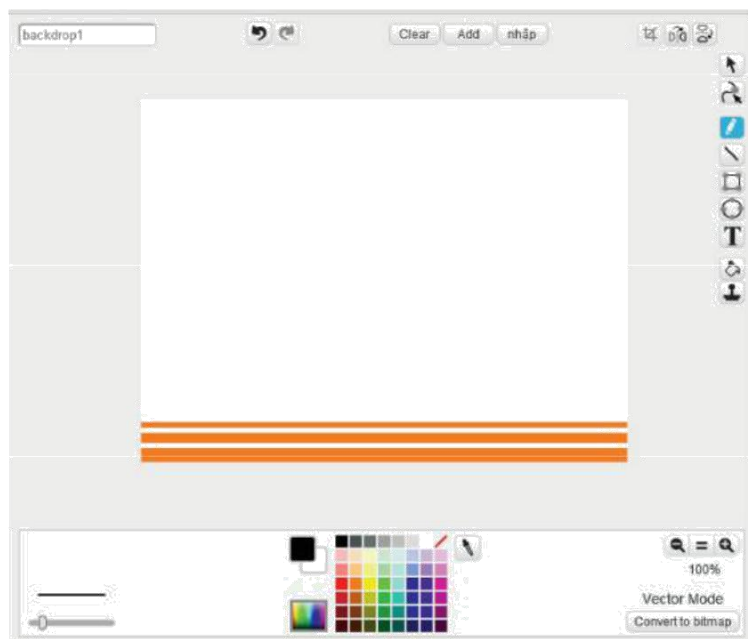
Khi bắt đầu vẽ, ta vào khu thiết kế phong nền bằng cách nhấp chuột vào phong nền sân khấu.



Hình 2.1.2a: Nhấp chuột vào phong nền sân khấu

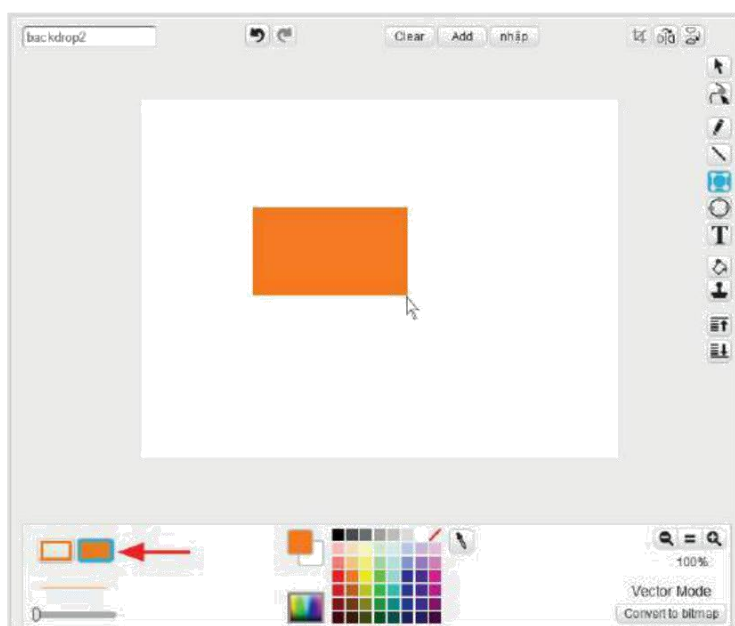
Sau đó nhấp chuột vào trang Backdrops. Các bạn nên chuyển sang chế độ Vector để chỉnh sửa hình ảnh dễ dàng hơn.





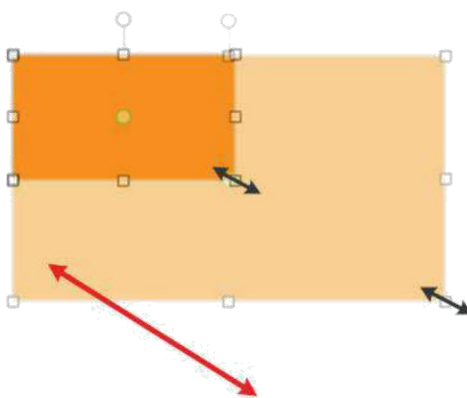
Hình 2.1.2. Khu thiết kế phong nền

Chúng ta vẽ ba hình chữ nhật ở phía bên dưới bằng công cụ **Vẽ hình chữ nhật** và chọn kiểu **Vẽ kèm màu nền**. Ta kéo thả chuột trên khu thiết kế để tạo hình chữ nhật.



Hình 2.1.3. Vẽ hình chữ nhật

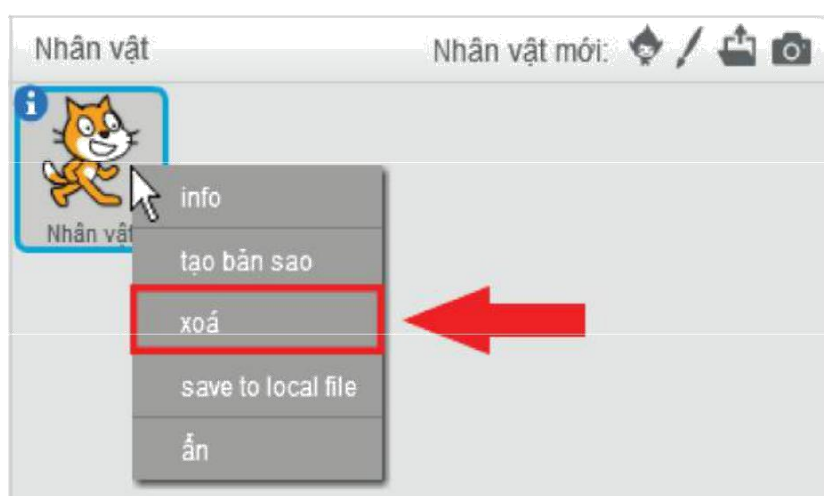
Với hai hình chữ nhật còn lại, ta thực hiện các bước tương tự. Ở chế độ Vector, ta có thể chỉnh sửa lại kích thước, vị trí và màu sắc cho các hình chữ nhật sau khi vẽ xong, còn với chế độ Bitmap thì chỉ có thể thay đổi được màu sắc.



Hình 2.1.4. Chỉnh sửa kích thước hình chữ nhật sau khi vẽ

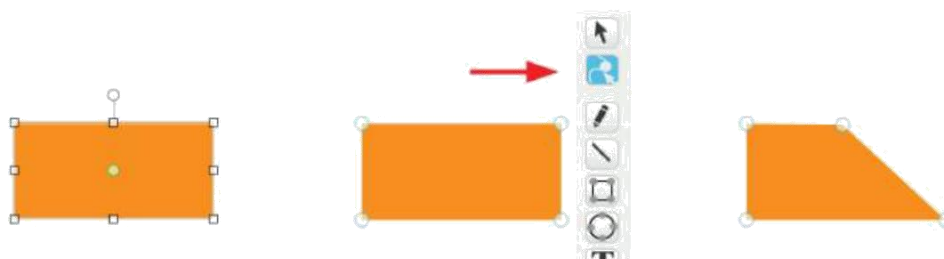
1.2. Chong chóng

Trước khi tạo các đối tượng cho chương trình, ta xóa đối tượng **Chú mèo** có sẵn bằng cách nhấp chuột phải vào hình ảnh chú mèo thu nhỏ, chọn xóa.



Hình 2.1.5. Xóa đối tượng nhân vật

Ở khung nhân vật, ra nhấp chuột vào nút **Vẽ mới** . Khi khu ngoại hình xuất hiện, các bạn chuyển sang chế độ Vector, sau đó bắt đầu vẽ cánh Chong chóng đầu tiên. Các bạn vẽ hình chữ nhật, chọn công cụ **Chỉnh hình** rồi kéo một góc của hình chữ nhật vào.



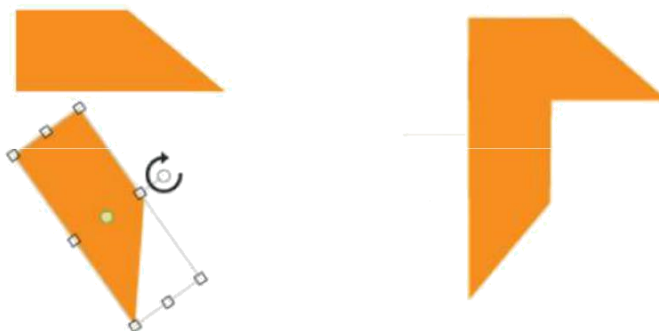
Hình 2.1.6. Vẽ cánh Chong chóng

Để vẽ cánh tiếp theo, ta nháy chuột vào công cụ **Tạo bản sao** , sau đó nháy chuột vào cánh chong chóng và kéo ra.



Hình 2.1.7. Tạo bản sao cho cánh chong chóng

Tiếp theo xoay cánh thứ hai vuông góc với cánh thứ nhất và kéo vào vị trí thích hợp.



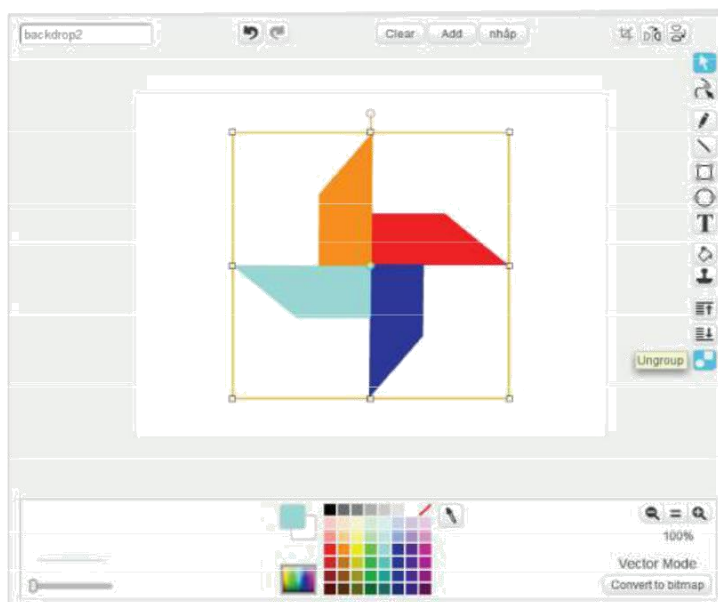
Hình 2.1.8. Vẽ cánh thứ 2

Làm tương tự với hai cánh chong chóng còn lại. Sau khi vẽ xong, các bạn đổ màu tùy ý cho chong chóng bằng công cụ **Đổ màu** .



Hình 2.1.9. Đổ màu các cánh Chong chóng

Cuối cùng, chúng ta gộp các cánh lại thành một hình hoàn chỉnh bằng cách quét chọn 4 cánh, sau đó dùng công cụ **Gộp nhóm** .



Hình 2.1.10. Chong chóng hoàn chỉnh



Tóm tắt lí thuyết và bài tập thực hành

Tạo hình nền và vẽ chong chóng

Vẽ chong chóng trên hình nền

daylaptrinh.net

Đối tượng

Cách vẽ

Bài tập

Tự sáng tạo một chiếc chong chóng có hình dáng và màu sắc khác.



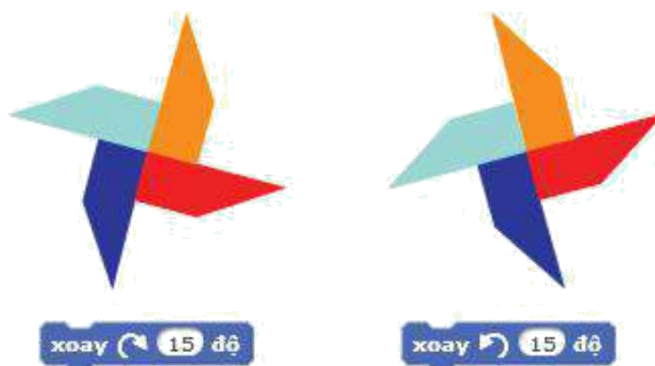
BÀI 2

LẬP TRÌNH CHONG CHÓNG QUAY

Sau khi đã tạo được Chong chóng, các bạn chuyển sang trang **Scripts** để lập trình cho chong chóng quay.

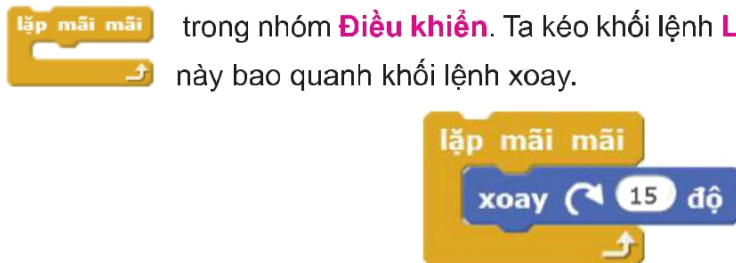
2.1. Khối lệnh lặp và xoay

Để lập trình cho Chong chóng quay, các bạn sử dụng khối lệnh xoay sang phải 15 độ  hoặc xoay sang trái 15 độ . Hai khối lệnh này sẽ giúp đối tượng xoay theo chiều sang trái và phải tương ứng với độ lớn góc xoay có thể được thay đổi tùy ý.



Hình 2.2.1. Khối lệnh xoay

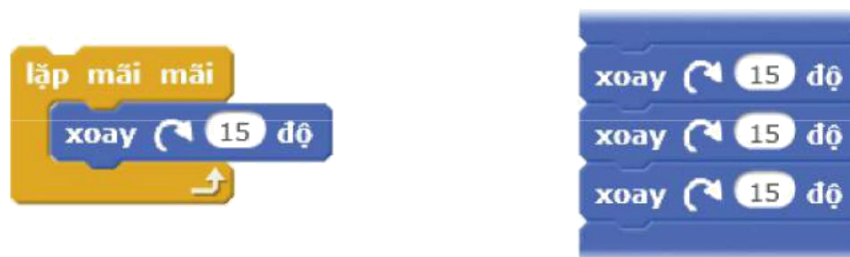
Ta kéo khối lệnh sang khu lập trình và nhấp chuột lên khối lệnh, khối lệnh sẽ được chạy. Tuy nhiên, Chong chóng chỉ xoay mỗi lần chúng ta chạy khối lệnh. Nếu muốn Chong chóng xoay liên tục, ta sử dụng khối lệnh lặp để khối lệnh xoay được lặp đi lặp lại. Trong trường hợp này chúng ta sử dụng khối lệnh **Lặp mãi mãi** trong nhóm **Điều khiển**. Ta kéo khối lệnh **Lặp mãi mãi** ra sao cho khối lệnh này bao quanh khối lệnh xoay.



Hình 2.2.2. Khối lệnh xoay và khối lệnh lặp

Trong lập trình gọi đây là một vòng lặp, những khối lệnh ở bên trong vòng lặp (được bao quanh bởi khối lệnh lặp) sẽ được thực hiện lặp đi lặp lại nhiều lần. Ở đây chúng ta sử dụng khối lệnh **Lặp mãi mãi** nên khối lệnh xoay ở bên trong sẽ được thực hiện lặp đi

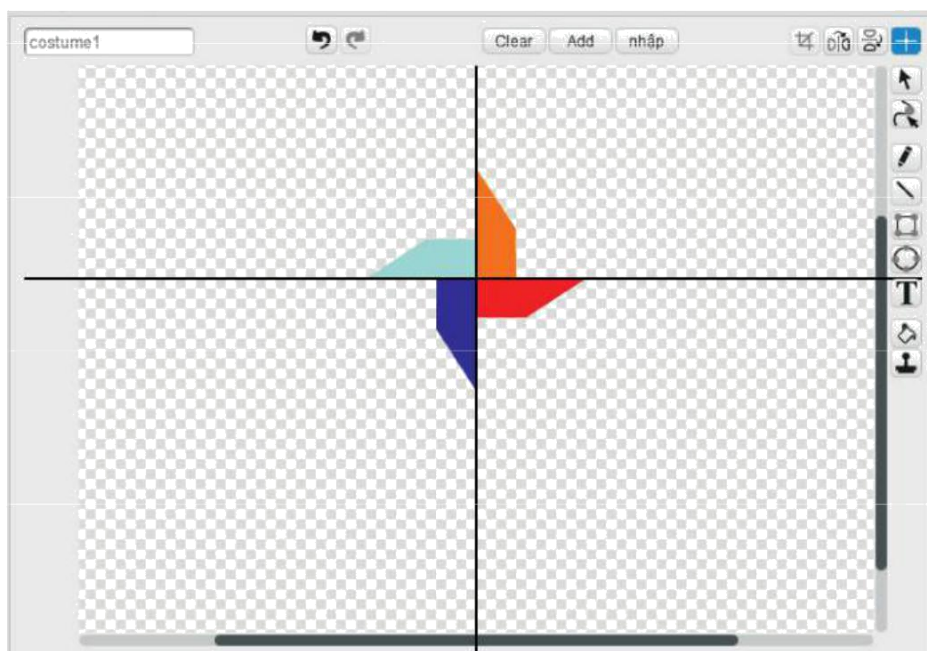
lặp lại mãi mãi cho đến khi nút **Dừng lại**  được chọn, và kết quả là Chong chóng sẽ quay liên tục không ngừng.



Hình 2.2.3. Hoạt động của vòng lặp

2.2. Điều chỉnh tâm của nhân vật

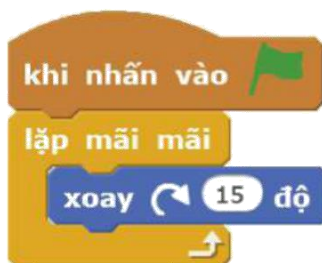
Khi khối lệnh được chạy, Chong chóng đã quay được nhưng đang quay quanh một điểm nào đó không phải ở chính giữa Chong chóng. Khi quay, đối tượng sẽ quay quanh tâm của nó, vậy nên để Chong chóng quay quanh điểm chính giữa của nó, cần đặt lại tâm của Chong chóng vào điểm chính giữa. Ta chuyển sang khu thiết kế (Costume), chọn nút **Điều chỉnh tâm**  và đặt lại tâm cho Chong chóng.



Hình 2.2.4. Chỉnh lại tâm cho Chong chóng

Cuối cùng, để khối lệnh chạy khi ta nhấp chuột vào lá cờ xanh, các bạn sử dụng khối lệnh **Khi nhấn vào lá cờ xanh**  và ghép lên trên vòng lặp.

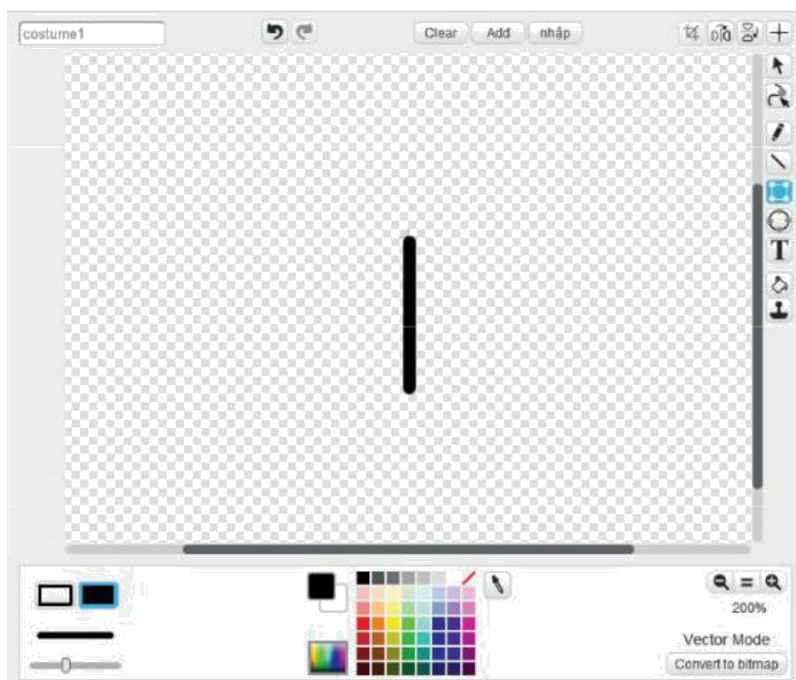




Hình 2.2.5. Đoạn khối lệnh hoàn chỉnh

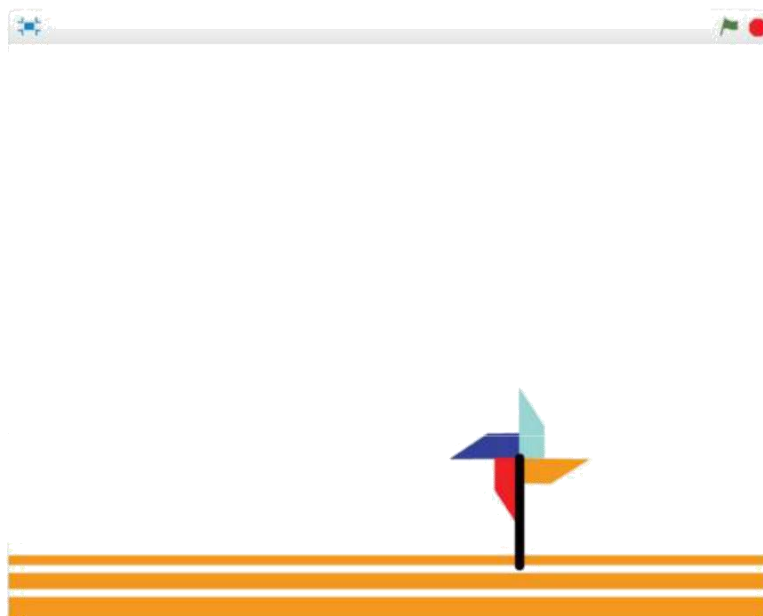
2.3. Đối tượng que

Trong chương trình, Chong chóng được gắn vào một chiếc Que. Nếu vẽ Que trực tiếp lên hình nền thì sẽ không thể di chuyển Que trong trường hợp muốn Chong chóng và Que thay đổi vị trí, còn nếu vẽ Que trực tiếp vào hình ảnh nhân vật của Chong chóng thì khi Chong chóng quay chiếc Que sẽ quay theo. Chính vì vậy, Que là một đối tượng tách riêng ra khỏi Chong chóng. Ta tạo nhân vật mới bằng cách dùng công cụ **Vẽ mới** , sau đó vẽ Que bằng hình chữ nhật.



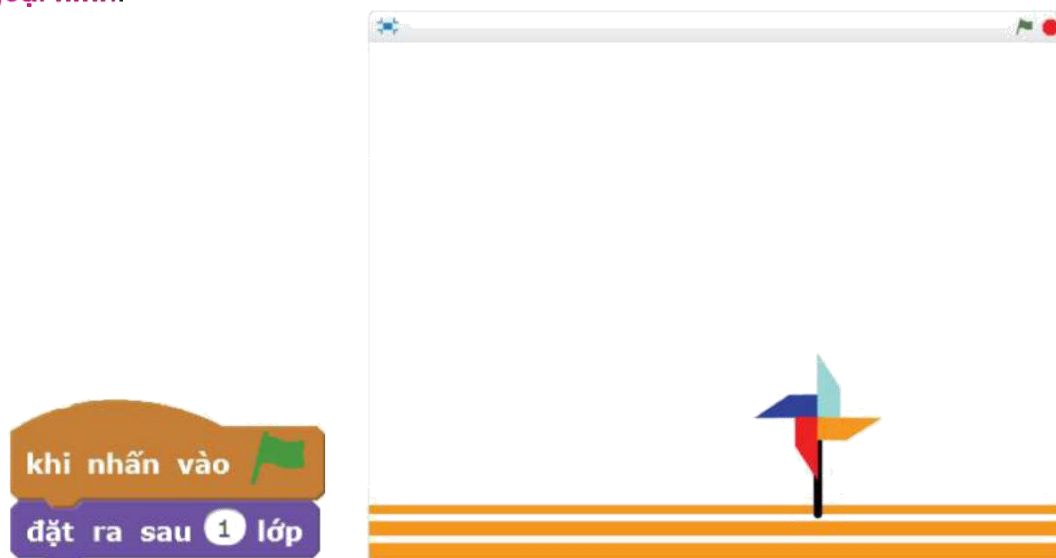
Hình 2.2.6. Vẽ đối tượng Que

Sau khi chỉnh tâm cho Que, các bạn kéo Que đến vị trí chong chóng sẽ thấy Que đang nằm phía trước và che khuất Chong chóng.



Hình 2.2.7. Vị trí của que đứng trước chong chóng trên sân khấu

Các nhân vật trên sân khấu đứng trước hay đứng sau là do được sắp xếp theo các lớp từ trước đến sau và mỗi nhân vật ở một lớp khác nhau. Để đặt Que ra phía sau Chong chóng, ta sử dụng lệnh **Đặt ra sau m lớp** đặt ra sau 1 lớp bên nhóm **Ngoại hình**.



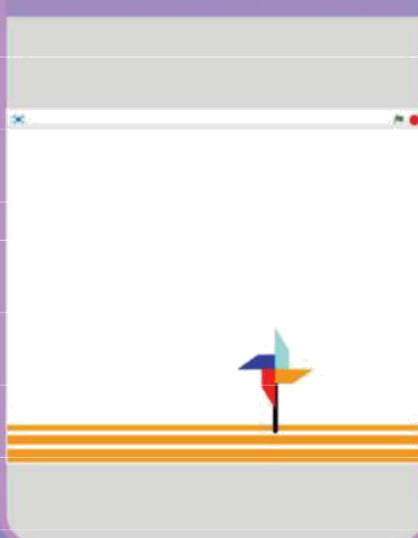
Hình 2.2.8. Đặt vị trí của Que ra sau Chong chóng



Tóm tắt lí thuyết và bài tập thực hành

Lập trình cho Chong chóng quay

Khối lệnh cho Chong chóng quay



daylaptrinh.net

Đối tượng



Chong chóng quay

Khối lệnh



Khối lệnh Chong chóng quay

Bài tập

1. Lập trình để Chong chóng quay với tốc độ nhanh hơn và chậm hơn.
2. Lập trình để đặt Chong chóng ra trước thay vì đặt Que ra sau.



BÀI 3

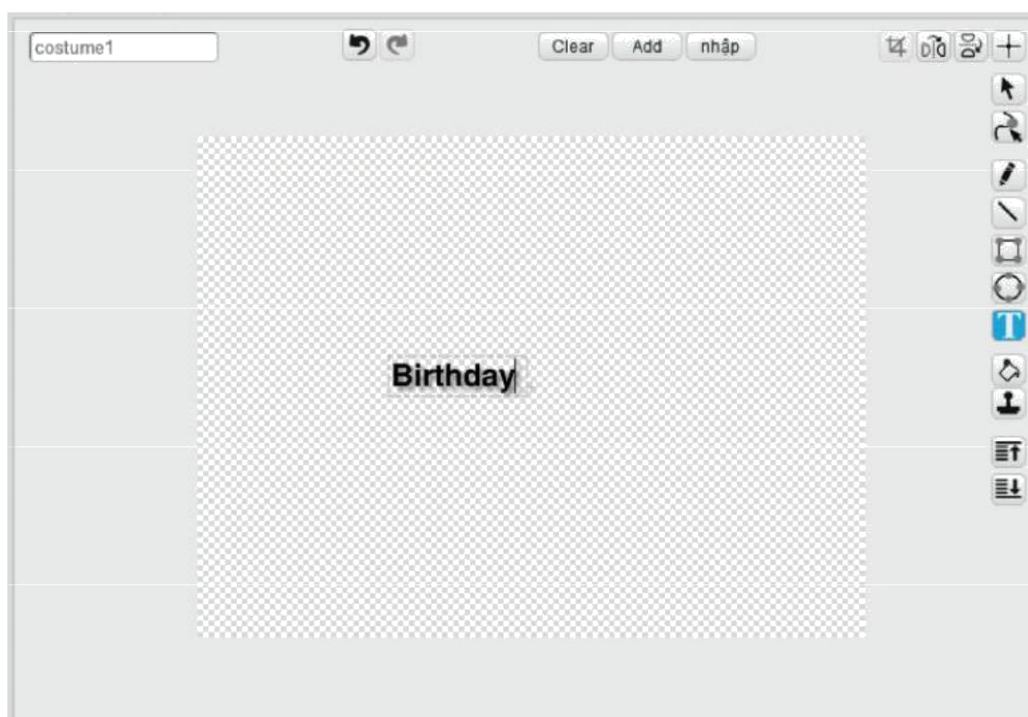
LẬP TRÌNH ĐỐI TƯỢNG CHỮ

Ở bài học này, các bạn sẽ được học cách thêm chữ Birthday và lập trình thay đổi hiệu ứng cho chữ ở trên thiệp.

3.1. Thêm và chỉnh sửa chữ

Chữ Birthday cũng là một đối tượng. Ta nháy vào nút **Vẽ mới**  trong danh sách nhân vật để thêm một nhân vật mới, trong khu thiết kế cho nhân vật mới, chúng ta sẽ thiết kế chữ Birthday.

Sau khi chuyển sang chế độ Vector, để thêm dòng chữ trong khu thiết kế, các bạn nháy vào nút **Thêm đoạn văn bản**  rồi nháy chuột vào vị trí bất kì trên khu thiết kế và gõ chữ "Birthday".



Hình 2.3.1. Thêm chữ cho thiệp

Thay đổi kích thước chữ bằng cách nháy vào **Nút chọn** , sau đó nháy vào chữ Birthday để chọn. Sau khi chọn, khung viền bao quanh chữ sẽ xuất hiện, nháy vào góc của khung viền và kéo khung viền to ra hoặc thu nhỏ lại.



Hình 2.3.2: Thay đổi kích thước chữ

Để đổi màu cho chữ, các bạn chọn màu ở bảng màu rồi sử dụng công cụ **Đổ màu** , hoặc chỉ cần nhấp nút **Chọn** , sau đó chọn chữ và chọn màu ở bảng màu là màu chữ sẽ tự động thay đổi theo.



Hình 2.3.3: Thay đổi màu

Để thay đổi nội dung hoặc phông chữ, nhấp vào nút **T**, sau đó nhấp vào chữ muốn chỉnh sửa. Lúc này, ta có thể thay đổi nội dung của chữ hoặc thay đổi phông chữ bằng cách chọn phông chữ mình muốn trong danh sách các phông chữ trong vùng tùy chọn.



Hình 2.3.4. Thay đổi phông chữ

3.2. Hiệu ứng

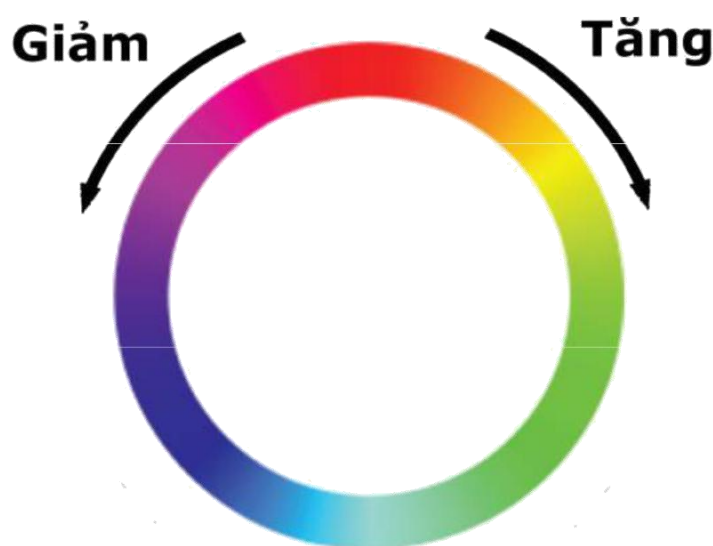
Việc thay đổi hiệu ứng của chữ được điều khiển bằng khối lệnh **Thay đổi hiệu ứng** ở nhóm **Ngoại hình**. Khối lệnh gồm loại hiệu ứng và lượng thay đổi hiệu ứng dưới dạng số. Trong khối lệnh hỗ trợ thay đổi nhiều hiệu ứng khác nhau.





Hình 2.3.5: Khối lệnh thay đổi hiệu ứng

Trong tấm thiệp, chữ sẽ luôn đổi màu, vì thế các bạn cần chọn hiệu ứng **Màu** trong khối lệnh. Màu chữ thay đổi phụ thuộc vào lượng giá trị trong khối lệnh. Lượng giá trị thay đổi là số dương (tăng) thì màu sẽ thay đổi theo chiều dương của k m đồng hồ, còn nếu lượng giá trị thay đổi là số âm (giảm) thì màu sẽ thay đổi theo chiều ngược lại, như được mô tả như trong Hình 2.3.6.



Hình 2.3.6. Màu thay đổi dựa trên lượng giá trị thay đổi

Lượng giá trị thay đổi của hiệu ứng màu có chu kì 200, tức là khi lượng giá trị thay đổi đạt đến một lượng bằng 200 thì hiệu ứng màu sẽ quay trở lại là màu gốc ban đầu. Với màu gốc là màu thì sự thay đổi hiệu ứng màu với giá trị dương từ 0 đến 200 được minh họa như trong Hình 2.3.7.



Hình 2.3.7. Hiệu ứng màu thay đổi theo chu kì giá trị hiệu ứng là 200



Để chữ có thể đổi màu liên tục, khối lệnh **Thay đổi hiệu ứng màu** được thực hiện lặp đi lặp lại nhiều lần, khối lệnh **Lặp mãi mãi** được sử dụng để chữ đổi màu cho đến khi dừng chương trình. Lúc này chữ đã đổi màu được liên tục nhưng rất nhanh. Ghép khối lệnh **đợi 1 giây** vào trong vòng lặp để sau mỗi lần đổi màu, chữ sẽ đợi thêm 1 giây rồi mới đổi màu tiếp. Và cuối cùng là khối lệnh **Kích hoạt** để khi bắt đầu chương trình (khi nháy vào lá cờ xanh) thì chữ sẽ bắt đầu đổi màu.



Hình 2.3.8. Đoạn khối lệnh hoàn chỉnh

óm tắt lí thuyết và ài tập thực hành

Chữ và Hiệu ứng chữ

Viết chữ và tạo hiệu ứng cho chữ

Birthday

Birthday

Birthday

daylaptrinh.net

Cách vẽ

Birthday Birthday

Birthday

Viết chữ, đổ màu và đổi font chữ

khối lệnh đổi hiệu ứng cho chữ

Công cụ

Vẽ mới Đổ màu Nút chọn Thêm văn bản

Bài tập

- Thực hành sử dụng phòng chữ khác cho chữ.
- Lập trình để thay đổi hiệu ứng mờ ảo cho chữ.

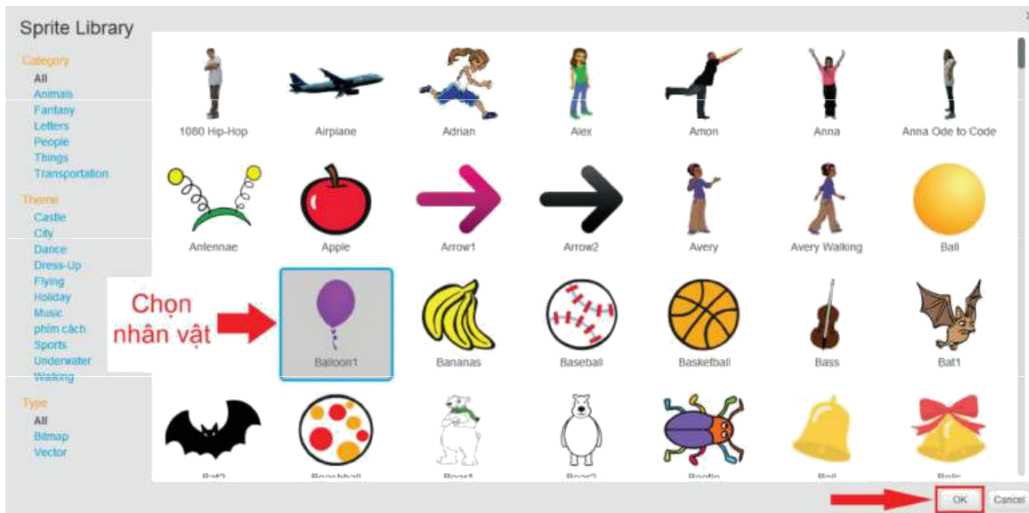
BÀI 4

LẬP TRÌNH ĐỐI TƯỢNG BÓNG BAY

Trong bài học này, chúng ta sẽ thêm đối tượng Bóng bay và lập trình để quả bóng có thể bay.

4.1. Thêm nhân vật Bóng bay

Để thêm quả bóng bay vào chương trình, ta chỉ cần nhấp nút  ở khung nhân vật để có thể chọn quả bóng bay từ thư viện có sẵn. Sau khi chọn nhấp nút  để hoàn tất.



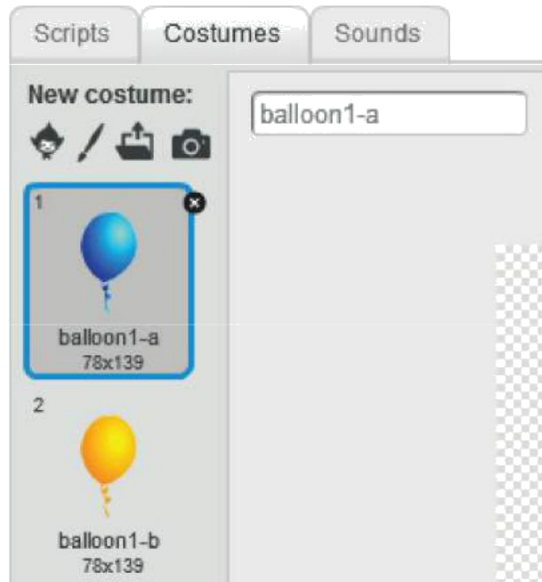
Hình 2.4.1. Thêm bóng bay từ thư viện

Tên mặc định của nhân vật khá khó đọc, ta nên thay đổi lại tên để dễ dàng tìm kiếm và không bị nhầm lẫn. Ở danh sách các đối tượng, ta nhấp vào biểu tượng  ở góc trên bên trái của nhân vật trong khung nhân vật để chỉnh sửa lại tên.



Hình 2.4.2. Đổi tên cho nhân vật

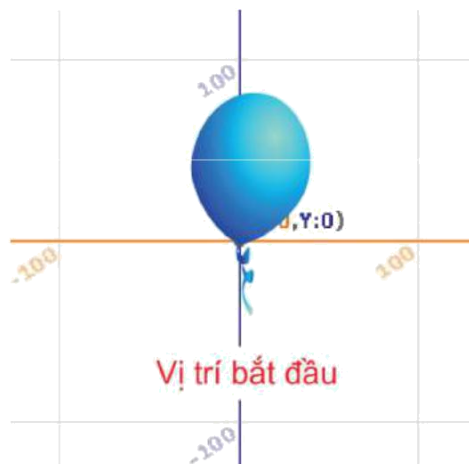
Nhân vật Bóng bay có sẵn ba hình dạng với màu sắc khác nhau, ta thay đổi hình ảnh của nhân vật tùy theo sở thích.



Hình 2.4.3. Đổi hình dạng Bóng bay

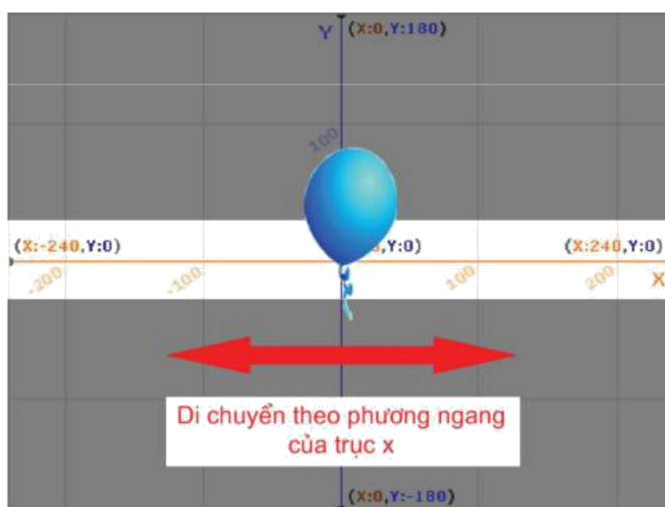
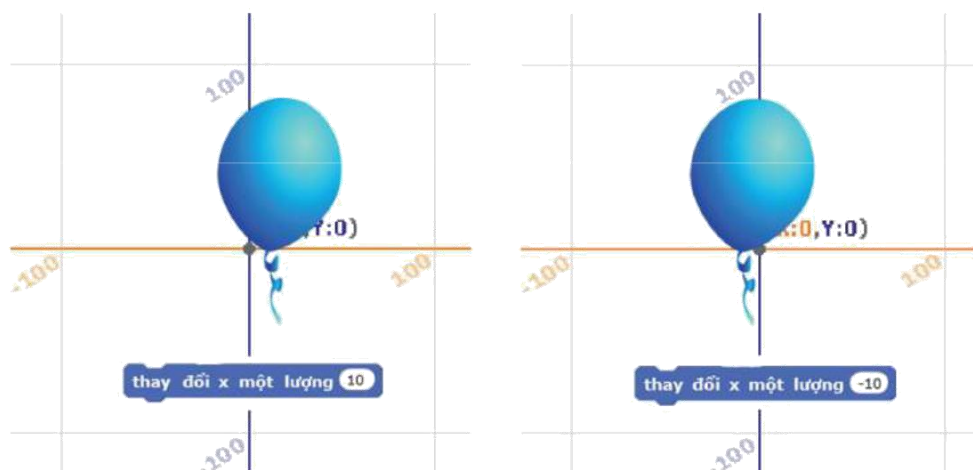
4.2. Lập trình Bóng ba

Nhân vật Bóng bay trong tấm thiệp cần được lập trình để có thể bay từ dưới lên trên. Trước tiên, chúng ta sẽ đi tìm hiểu về sự di chuyển của nhân vật trên sân khấu bằng cách thay đổi tọa độ x và y . Ở thẻ **Scripts**, các bạn chạy lần lượt các khối lệnh **thay đổi x một lượng** và **thay đổi y một lượng** với hai giá trị là 10 và -10, sau đó theo dõi kết quả.



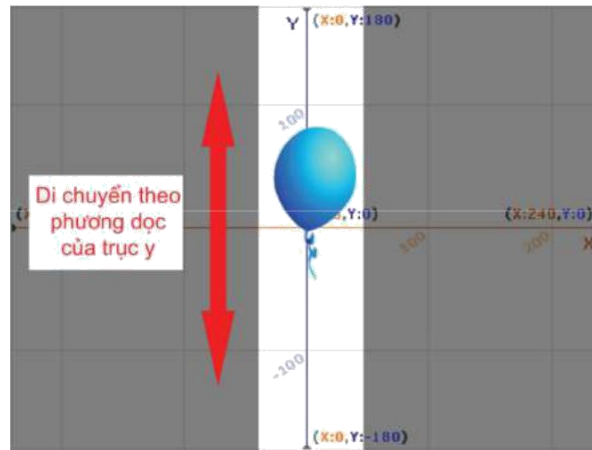
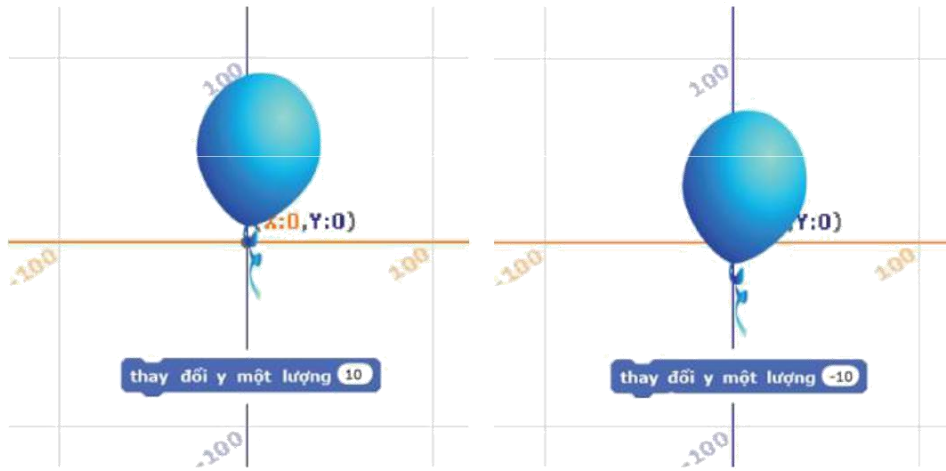
Hình 2.4.4. Vị trí bắt đầu của quả bóng

Khi chạy khối lệnh **thay đổi x một lượng 10**, quả bóng di chuyển sang phải 10 bước (10 pixel), còn chạy khối lệnh **thay đổi x một lượng -10** sẽ khiến quả bóng di chuyển sang trái 10 bước. Như vậy, việc thay đổi giá trị x của nhân vật sẽ khiến nhân vật di chuyển theo phương ngang, tức là phương theo trục x .



Hình 2.4.5. Thay đổi vị trí x một lượng nhất định

Khi chạy khối lệnh **thay đổi y một lượng 10**, quả bóng sẽ di chuyển lên trên 10 bước, còn nếu lượng thay đổi là -10 như trong khối lệnh **thay đổi y một lượng -10** thì quả bóng sẽ di chuyển xuống dưới 10 bước. Vậy việc thay đổi giá trị y của nhân vật sẽ khiến nhân vật di chuyển theo phương thẳng đứng, tức là theo phương của trục y .



Hình 2.4.6. Thay đổi vị trí y một lượng nhất định

Quả bóng cần phải di chuyển theo phương thẳng đứng với chiều hướng từ dưới lên trên, vì vậy các bạn sử dụng khối lệnh **thay đổi y một lượng** . Ở Chương I, chúng ta đã được biết sân khấu có chiều dọc là 360 bước, vậy nên để di chuyển hết chiều này thì bóng bay cũng phải di chuyển hết 360 bước. Trong trường hợp này, chúng ta sẽ điều khiển quả bóng di chuyển 36 lần, mỗi lần 10 bước bằng cách thực hiện khối lệnh **thay đổi y một lượng** lặp đi lặp lại 36 lần.



Hình 2.4.7. Đoạn khối lệnh di chuyển của Bóng bay

Ta có thể điều chỉnh số lần lặp để thay đổi độ dài quãng đường di chuyển cũng như điều chỉnh lượng giá trị y để thay đổi tốc độ di chuyển của quả bóng nhanh hay chậm tùy ý.

Cuối cùng, để đoạn khối lệnh chạy khi nháy vào nút lệnh lá cờ xanh , ta sử dụng khối lệnh **Khi nhấn vào lá cờ xanh**  và ghép lên trên khối lệnh.



Hình 2.4.8. Đoạn khối lệnh hoàn chỉnh



Tóm tắt lí thuyết và bài tập thực hành

Lập trình cho Bóng bay

Thêm bóng bay và lập trình cho Bóng bay

daylaptrinh.net

Hướng di chuyển

Khối lệnh

Bài tập

1. Lập trình để Bóng bay bay chậm hơn.
2. Lập trình để Bóng bay di chuyển theo phương ngang.

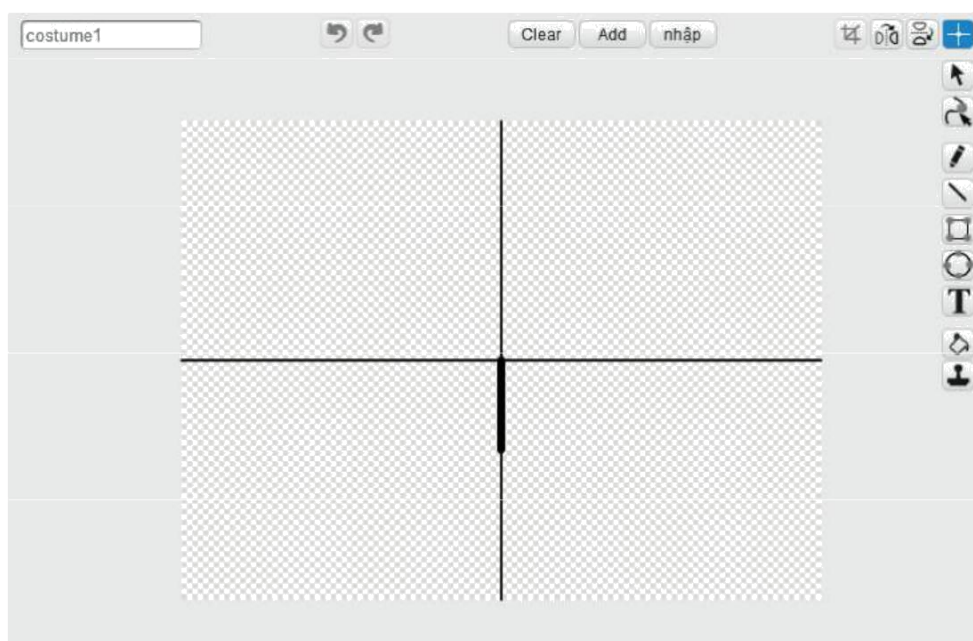
BÀI 5

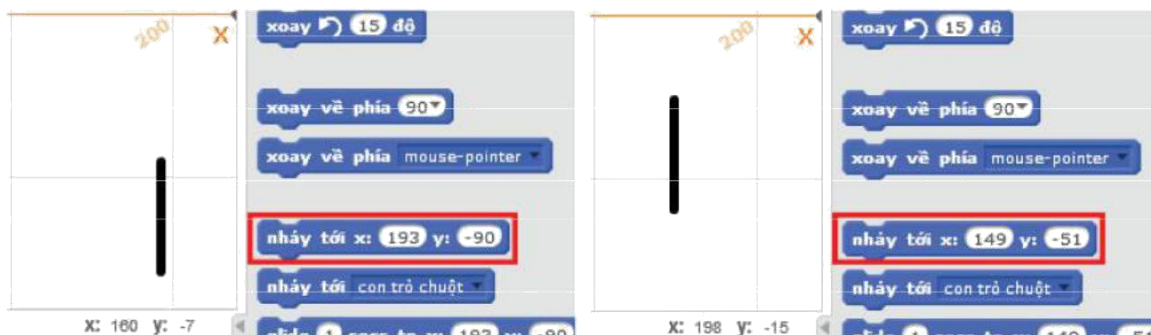
VỊ TRÍ BẮT ĐẦU CỦA CÁC ĐỐI TƯỢNG

Trong rất nhiều trường hợp, chúng ta muốn khi bắt đầu chương trình (nháy vào nút lệnh lá cờ xanh) thì đối tượng sẽ tự động nhảy đến một vị trí cố định trên sân khấu. Bài học này hướng dẫn các bạn lập trình để thực hiện điều đó.

5.1. Nhảy tới vị trí nhất định

Để lập trình cho nhân vật nhảy tới một vị trí nhất định trên sân khấu, ta sử dụng khối lệnh **Nhảy tới tọa độ x, y** **nhảy tới $x: 0 y: 0$** trong nhóm **Chuển động**. Khối lệnh này sẽ điều khiển nhân vật chuyển tới vị trí có tọa độ x, y trong khối lệnh ngay lập tức. Tọa độ của nhân vật chính là vị trí tọa độ của tâm nhân vật.





Hình 2.5.2: Khối lệnh **Nhảy tới tọa độ** tự động thay đổi giá trị x, y

Như vậy, khi ta kéo nhân vật đến vị trí mong muốn thì khối lệnh **nhảy tới x: y:** sẽ thay đổi giá trị x, y thành tọa độ của vị trí mong muốn đó. Kéo khối lệnh sang khu lập trình, thêm khối lệnh **khi nhấn vào** lên bên trên khối lệnh để khi bắt đầu chương trình (nháy vào nút lệnh lá cờ xanh) nhân vật sẽ nhảy đến vị trí mong muốn ngay lập tức.



Hình 2.5.3. Đoạn khối lệnh hoàn chỉnh



Tóm tắt lý thuyết và bài tập thực hành

Vị trí bắt đầu của các đối tượng

Nhảy tới vị trí nhất định

daylaptrinh.net

Chỉnh tâm

Cách dùng

Bài tập

Lập trình để thêm vị trí bắt đầu cho Chong chóng và chữ Birthday.



BÀI 6 BÓNG BAY BAY LÊN TỪ VỊ TRÍ NGẪU NHIÊN

Sau khi học Bài 4, các bạn đã biết cách lập trình để quả bóng có thể bay lên, tuy nhiên sau khi bay lên quả bóng sẽ đứng yên ở đó. Bài học này sẽ hướng dẫn các bạn lập trình để điều khiển quả bóng lặp đi lặp lại việc bay lên và mỗi lần bay lên quả bóng lại xuất phát từ một vị trí khác.

6.1. Vòng lặp lồng nhau

Đoạn khối lệnh điều khiển quả bóng bay lên gồm hai thành phần: khối lệnh nhảy tới tọa độ **nhảy tới x: 10 y: -191** điều khiển vị trí quả bóng và vòng lặp thay đổi giá trị **thay đổi y một lượng 10** điều khiển quả bóng bay lên.



Hình 2.6.1. Điều khiển quả bóng bay lên

Chúng ta sử dụng khối lệnh **Lặp mãi mãi** để lập trình việc nhảy tới vị trí mới và điều khiển quả bóng bay lên lặp đi lặp lại.

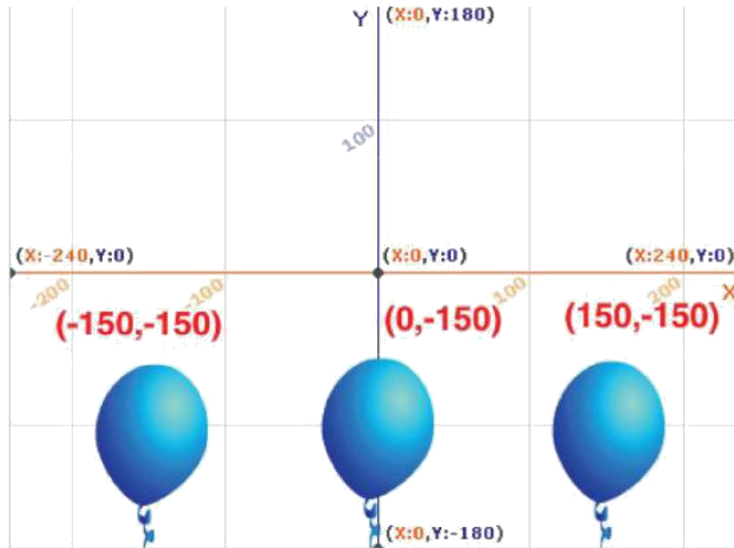


Hình 2.6.2. Đoạn khối lệnh lặp lồng nhau

Việc nhảy đến vị trí và bay lên được lặp đi lặp lại mãi mãi cho đến khi ta nhấp vào biểu tượng **Dừng**.

6.2. Lấy giá trị ngẫu nhiên

Khi vị trí x thay đổi, vị trí theo phương ngang của nhân vật sẽ bị thay đổi.



Hình 2.6.3. Thay đổi vị trí x

Chính vì vậy, để thay đổi vị trí bay lên của quả bóng, các bạn cần thay đổi vị trí x sau mỗi lần lặp (mỗi lần bay lên). Chúng ta thay đổi vị trí x một cách ngẫu nhiên bằng khối lệnh **Lấy ngẫu nhiên** lấy ngẫu nhiên trong khoảng tới . Khối lệnh này sẽ chứa một giá trị được lấy ngẫu nhiên trong khoảng từ số thứ nhất đến số thứ hai. Vị trí theo chiều ngang (trục x) của đối tượng Bóng bay bị giới hạn trong khu vực sân khấu với x có giá trị từ -240 đến 240 . Chúng ta ghép khối lệnh đã được chỉnh sửa giá trị lấy ngẫu nhiên trong khoảng tới vào giá trị x của khối lệnh nhảy tới x : y : . Vậy là sau mỗi lần lặp, khối lệnh nhảy tới x : lấy ngẫu nhiên trong khoảng tới y : sẽ không còn khiến nhân vật nhảy đến một vị trí cố định nữa mà sẽ có vị trí theo chiều ngang khác nhau.



Hình 2.6.4. Đoạn khối lệnh hoàn chỉnh



Tóm tắt lý thuyết và bài tập thực hành

Bóng bay bay lên từ vị trí ngẫu nhiên

Vòng lặp lồng nhau và giá trị ngẫu nhiên

daylaptrinh.net

Vòng lặp lồng nhau

```

lặp mãi mãi
  nhảy tới x: 10 y: -191
  lặp lại 36 lần
    thay đổi y một lượng 10
        
```

Lấy giá trị ngẫu nhiên

```

khi nhấn vào
  lặp mãi mãi:
    nhảy tới x: lấy ngẫu nhiên trong khoảng -240 tới 240 y: -191
    lặp lại 36 lần
      thay đổi y một lượng 10
        
```

Bài tập

Lập trình để Bóng bay nhảy tới vị trí ngẫu nhiên ở nửa phần dưới của sân khấu.



BÀI 7

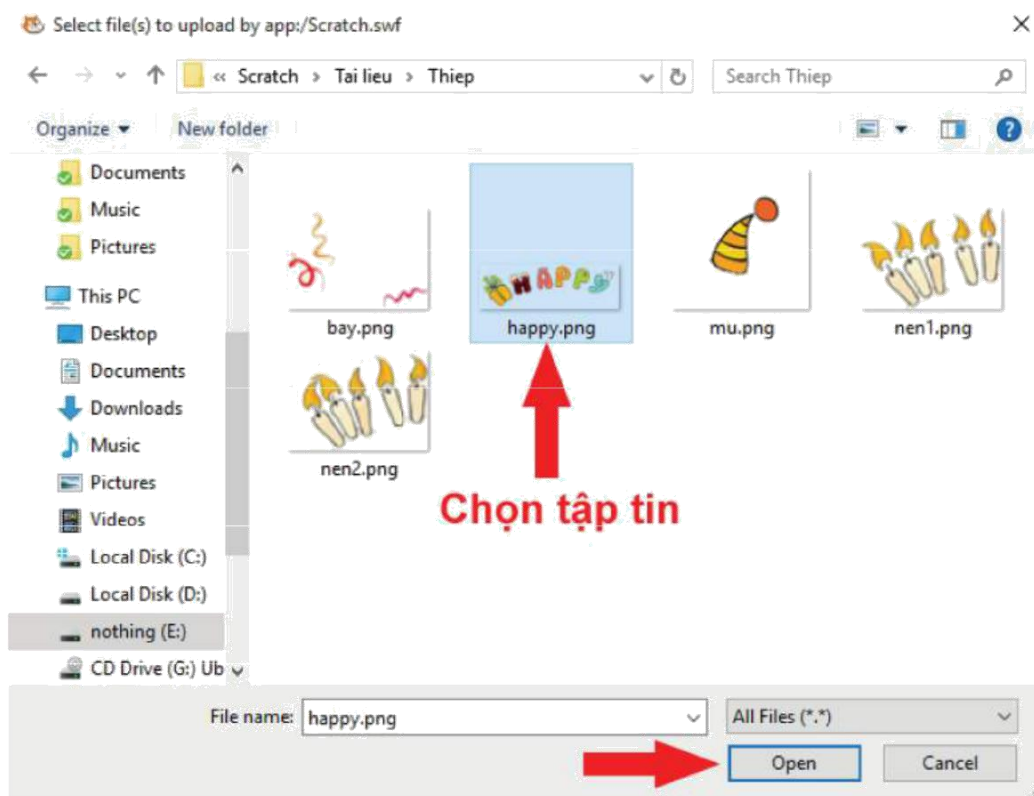
LẬP TRÌNH ĐỐI TƯỢNG NÉN

Trong bài học này, các bạn sẽ được hướng dẫn cách thêm nhân vật chữ Happy và nhân vật Nén từ thư mục trên máy tính, đồng thời lập trình nén có thể cháy.

7.1. Thêm nhân vật từ thư mục

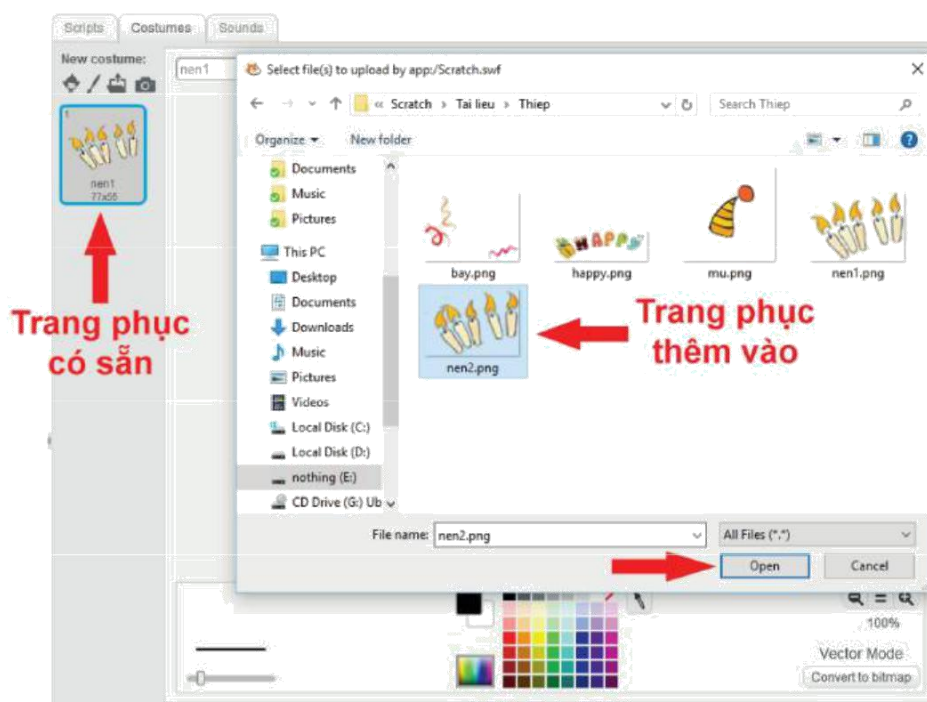
Đầu tiên, các bạn lên trang web www.daylaptrinh.net để tải tài liệu về máy tính.

Ở phần danh sách nhân vật, ta nhấn vào nút , cửa sổ mới sẽ hiện lên để chọn tập tin. Ta tìm đến thư mục tài liệu vừa tải về, chọn tập tin rồi nhấn Open.



Hình 2.7.1. Thêm đối tượng chữ Happy từ thư mục

Tiếp theo là thêm ngọn nến. Nén có hai trang phục, vậy nên sau khi thêm nhân vật nến thì ta cần thêm một trang phục nữa cho Nén. Ở khu ngoại hình của đối tượng nến ta nhấn vào nút  và chọn trang phục thứ hai của nến.



Hình 2.7.2. Thêm trang phục cho nến

7.2. Lập trình cho đối tượng nến

Nến được lập trình chuyển đổi giữa hai trang phục liên tục, khi đó trên sân khấu nến sẽ như đang bập bùng cháy. Để thay đổi trang phục, chúng ta sử dụng khối lệnh **Hình dạng tiếp theo** hình dạng tiếp theo trong nhóm **Ngoại hình**. Khi nháy chạy khối lệnh, nhân vật nến sẽ thay đổi hình ảnh của mình. Khi khối lệnh chạy liên tục sẽ thấy nến như đang cháy.



Hình 2.7.3. Hai trang phục của đối tượng Nến

Chúng ta sử dụng khối lệnh **Lặp mãi mãi** để nến luôn cháy, tuy nhiên khi chạy thử thì nến thay đổi trang phục rất nhanh và không đẹp. Các bạn ghép thêm khối lệnh đợi **đợi 1 giây** vào trong vòng lặp để nến cháy chậm và đẹp hơn. Chúng ta có thể chỉnh sửa thời gian đợi thành 0.5 giây hay 0.2 giây để thay đổi tốc độ cháy như



trong đoạn khối lệnh  . Thêm khối lệnh **Kích hoạt**  khi nhấn vào

và **Nhảy tới tọa độ**  để khi bắt đầu chương trình ngọn nến sẽ nhảy đến vị trí mong muốn.



Hình 2.7.4. Đoạn khối lệnh hoàn chỉnh



Tóm tắt lí thuyết và bài tập thực hành

Lập trình đối tượng Nến

Thêm đối tượng và lập trình



daylaptrinh.net

Hình dạng nến



Nến 1 Nến 2

Khối lệnh



Bài tập

Lập trình để thêm vị trí bắt đầu cho chữ Happy.

BÀI 8

ĐỐI TƯỢNG MŨ VÀ PHÁO

Bài học này sẽ hướng dẫn các bạn lập trình hai đối tượng Mũ và Pháo. Hai nhân vật này đều có chuyển động lên và xuống lặp đi lặp lại khá giống nhau.

8.1. Thêm nhân vật từ thư mục

Cách thêm Mũ và Pháo giống với nhân vật Nền.



Hình 2.8.1. Thêm nhân vật Mũ và nhân vật Pháo

8.2. Lập trình cho Mũ

Chiếc mũ sẽ chuyển động lên và xuống đều đặn. Qua bài 4, chúng ta đã biết muốn điều khiển đối tượng di chuyển theo phương thẳng đứng sẽ phải thay đổi vị trí y của nhân vật, vậy ta cần sử dụng hai khối lệnh thay đổi giá trị y

thay đổi y một lượng

. Trong hai khối lệnh này, một khối lệnh sẽ làm tăng giá trị y (lượng giá trị y thay đổi là số dương) để nhân vật chuyển động theo hướng từ dưới lên trên, và khối lệnh còn lại sẽ giảm giá trị y (lượng giá trị y thay đổi là số âm) để nhân vật chuyển động theo hướng từ trên xuống dưới.



Hình 2.8.2. Đoạn khối lệnh di chuyển của Mũ

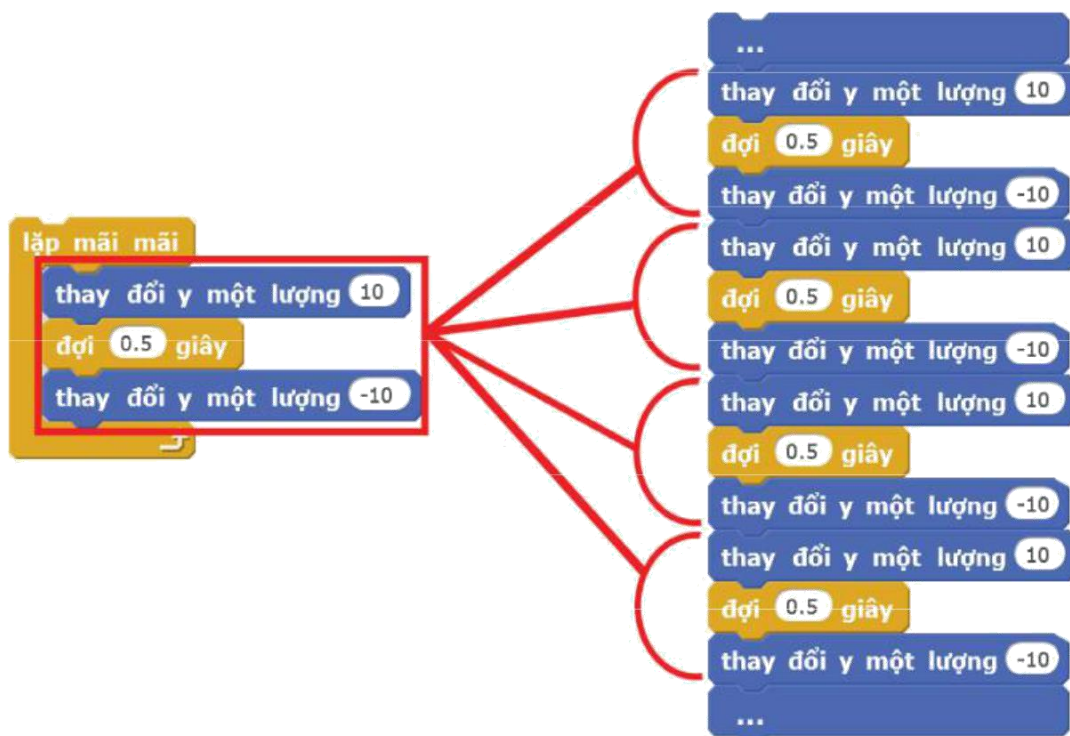


Khi chạy đoạn khối lệnh này, ta sẽ thấy nhân vật vẫn đứng im, vì các khối lệnh thực hiện rất nhanh, vừa di chuyển lên với khối lệnh **thay đổi y một lượng 10** nhân vật đã di chuyển xuống bằng khối lệnh **thay đổi y một lượng -10** ngay lập tức. Chính vì vậy, chúng ta ghép thêm khối lệnh đợi **đợi 1 giây** vào giữa hai khối lệnh trên để chuyển động được quan sát rõ ràng hơn. Ta có thể tùy chỉnh thời gian đợi để thay đổi tốc độ lên và xuống của chiếc mũ.



Hình 2.8.3. Thêm khối lệnh đợi giữa hai lần di chuyển

Để việc di chuyển lên và xuống này được thực hiện liên tục, ta ghép thêm vòng lặp bao quanh bên ngoài đoạn khối lệnh. Hoạt động của đoạn khối lệnh có thêm vòng lặp được mô tả ở Hình 2.8.2.



Hình 2.8.4. Hoạt động vòng lặp

Giữa các lần lặp với nhau không có khối lệnh đợi, sau khi khối lệnh **thay đổi y một lượng -10** được chạy, ngay lập tức khối lệnh **thay đổi y một lượng 10** cũng được thực hiện, chúng ta không kịp nhìn thấy chiếc mũ di chuyển xuống. Vậy nên chúng ta ghép thêm một khối lệnh **đợi 0.5 giây** nữa.



Hình 2.8.5. Thêm khối lệnh **Đợi**

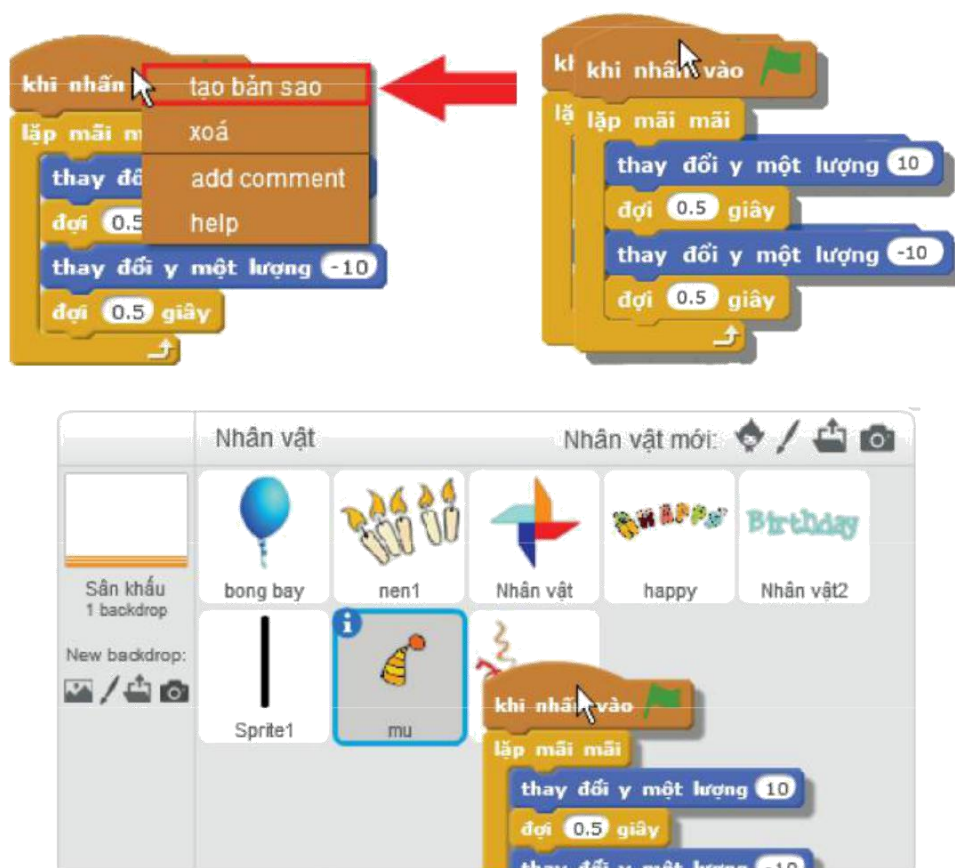
Vậy là đoạn khối lệnh cho việc di chuyển đã xong, cần nhớ thêm khối lệnh

Kích hoạt **khi nhấn vào** cho đo n khối lệnh của mình.

8.3. Lập trình cho Pháo

Pháo được lập trình di chuyển lên và xuống liên tục giống chiếc mũ, khi Mũ di chuyển lên thì Pháo di chuyển xuống và ngược lại. Ta sao chép đoạn khối lệnh của Mũ, sau đó kéo sang nhân vật Pháo bằng cách nháy nút phải chuột lên khối lệnh trên cùng

khi nhấn vào, chọn **Tạo bản sao**. Khối lệnh mới xuất hiện giống khối lệnh cũ được gắn vào con trỏ chuột mà không cần phải nhấn giữ chuột. Di chuyển con trỏ chuột và khối lệnh đến ảnh thu nhỏ của nhân vật Pháo và nháy chuột vào đó, khối lệnh sẽ được xuất hiện bên khu lập trình của nhân vật Pháo.

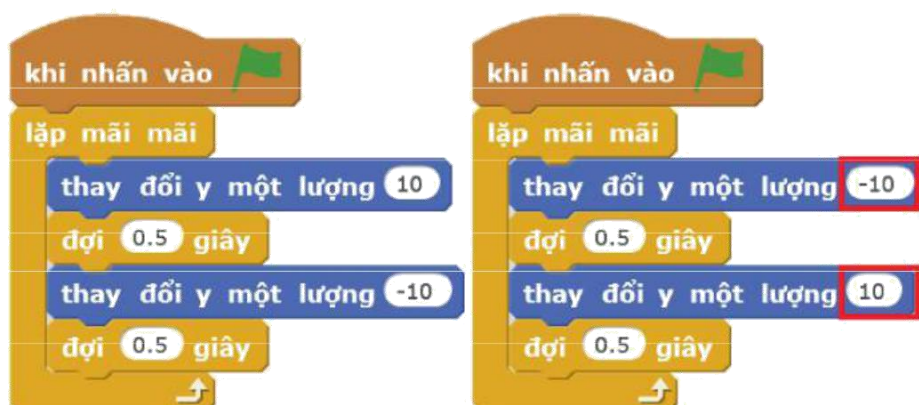


Hình 2.8.6. Sao chép khối lệnh



Hình 2.8.7. Kết quả

Khi chạy chương trình, Mũ và Pháo trên sân khấu có chuyển động giống hệt nhau vì chúng có đoạn khối lệnh điều khiển giống nhau. Để chuyển động của Pháo ngược với Mũ, các bạn đổi chỗ hai giá trị của hai khối lệnh thay đổi y thay đổi y một lượng trong đoạn khối lệnh của Pháo. Đối tượng Mũ có khối lệnh di chuyển đầu tiên là di chuyển lên, đối tượng Pháo có khối lệnh di chuyển đầu tiên là di chuyển xuống.



Hình 2.8.8. Đoạn khối lệnh hoàn chỉnh của Mũ và Pháo



Tóm tắt lí thuyết và bài tập thực hành

Đối tượng Mũ và Pháo

Lập trình cho Mũ và Pháo

daylaptrinh.net

Đối tượng

Mũ
 Pháo hoa

Khối lệnh

Bài tập

1. Lập trình để thêm vị trí bắt đầu cho Mũ và Pháo.
2. Lập trình để Mũ và Pháo chuyển động chậm hơn.

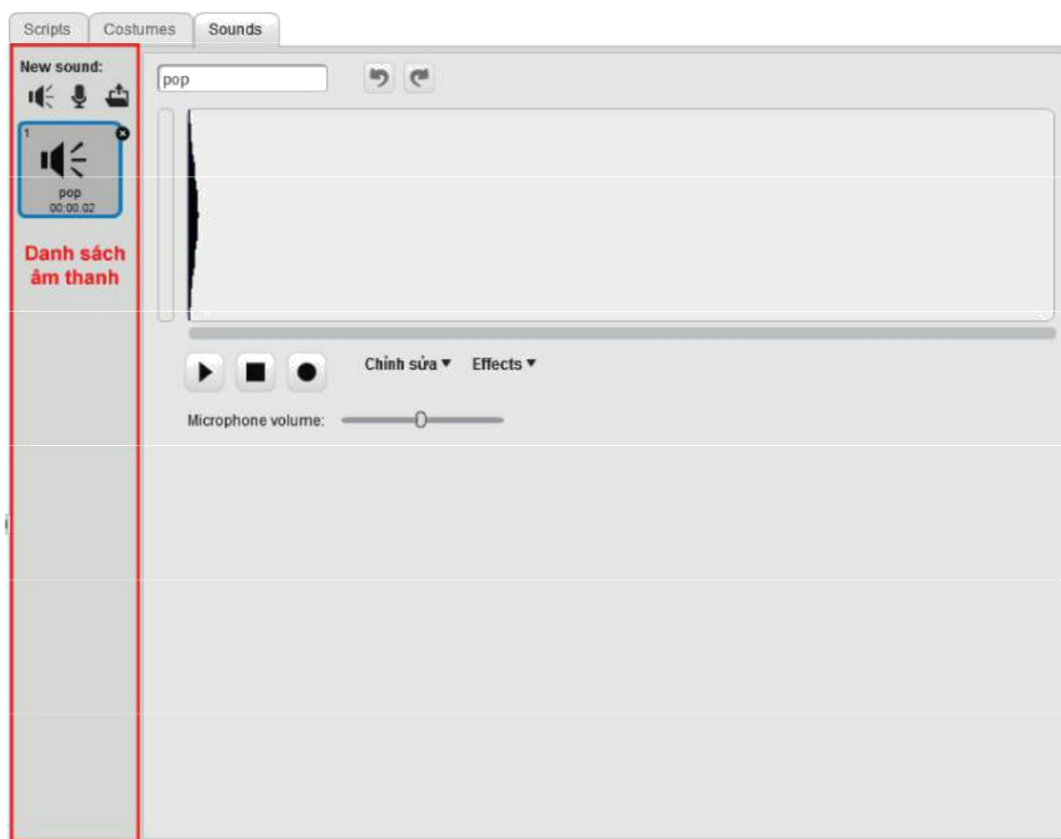
BÀI 9

ÂM THANH

Bài học này sẽ hướng dẫn các bạn lập trình để thêm âm thanh cho đối tượng và thực hành để thêm nhạc vào thiệp.

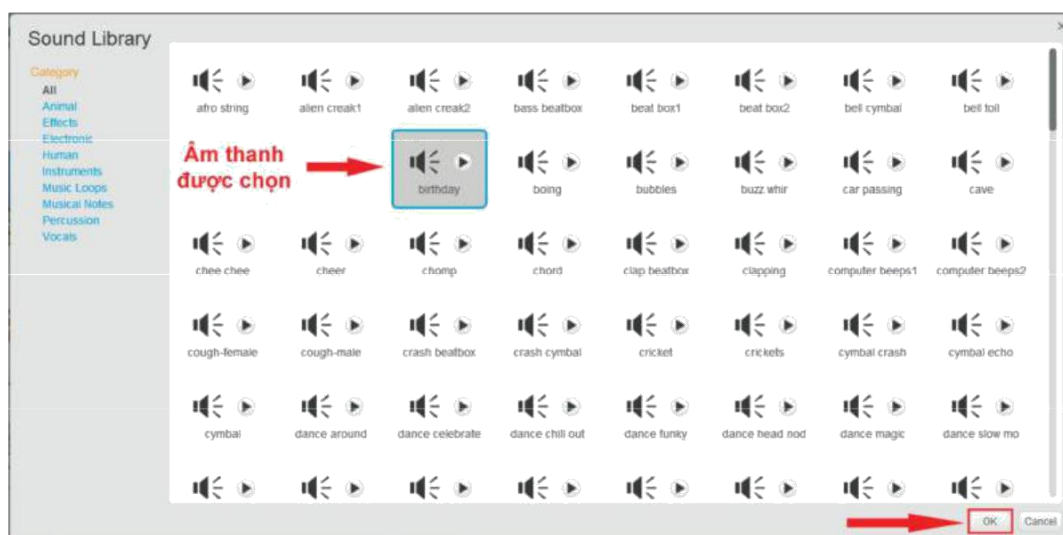
9.1. Thêm âm thanh

Mỗi nhân vật, đối tượng đều có danh sách âm thanh riêng ở **Khu âm thanh** (trang **Sounds**). Đây là nơi chứa tất cả âm thanh mà nhân vật có thể chơi. Mỗi nhân vật được thêm mới (từ thư viện, tạo mới, tải từ máy tính hay chụp ảnh) đều được tạo sẵn âm thanh mặc định.



Hình 2.9.1. Danh sách âm thanh ở **Khu âm thanh**

Âm thanh của nhân vật nào sẽ được thêm vào danh sách âm thanh của nhân vật đó, còn nhạc nền thì sẽ đưa vào danh sách âm thanh của sân khấu. Chương trình Thiệp sinh nhật có nhạc nền là bài Happy Birthday là âm thanh đã có sẵn trong thư viện âm thanh, chúng ta sẽ thêm âm thanh có sẵn trong thư viện bằng cách nhấp vào nút .



Hình 2.9.2. Thêm nhạc birthday cho thiệp

9.2. Lập trình

Trong Danh mục các nhóm lệnh, nhóm **Âm thanh** có hai khối lệnh dùng để chơi âm thanh là **chơi âm thanh birthday** và **chơi âm thanh birthday cho đến hết**. Cả hai khối lệnh đều dùng để chơi âm thanh có trong danh sách âm thanh của đối tượng sân khấu.



Hình 2.9.3. Hai khối lệnh chơi âm thanh

Âm thanh “birthday” chơi liên tục cho đến khi dừng chương trình, vì vậy ta

ghép khối lệnh **lặp mãi mãi** bao bên ngoài khối lệnh chơi âm thanh. Hai khối lệnh chơi âm thanh được ghép vào trong vòng lặp, khi chạy sẽ có hai kết quả khác nhau.

Đoạn khối lệnh **lặp mãi mãi** **chơi âm thanh birthday cho đến hết** sẽ chơi âm thanh Birthday, còn với đoạn

khối lệnh **lặp mãi mãi** **chơi âm thanh birthday** chúng ta sẽ không nghe thấy âm thanh nào được chơi.



Khi một âm thanh đang được chơi, nếu chạy lại khối lệnh chơi âm thanh đó thì âm thanh sẽ được chơi lại từ đầu. Khối lệnh **chơi âm thanh birthday cho đến hết** sẽ chơi âm thanh được chọn và đợi cho đến khi âm thanh đó được chơi hết mới thực hiện công việc các khối lệnh tiếp theo (trong trường hợp này là lặp lại việc chơi lại âm thanh birthday). Còn khối lệnh **chơi âm thanh birthday** sẽ chơi âm thanh và thực hiện các công việc tiếp theo luôn mà không chờ cho đến khi âm thanh được chơi hết. Ở trong vòng lặp, khối lệnh này sẽ liên tục được thực hiện, âm thanh liên tục được chơi lại từ đầu nên chúng ta sẽ không nghe thấy gì.

Sau khi biết được phải dùng khối lệnh nào cho vòng lặp, các bạn ghép thêm khối lệnh kích hoạt **khi nhấn vào** để nhạc sẽ được phát từ lúc bắt đầu chương trình.



Hình 2.9.4. Đoạn khối lệnh hoàn chỉnh



Tóm tắt lí thuyết và bài tập thực hành

Âm Thanh

Thêm âm thanh và lập trình

daylaptrinh.net

Công cụ

Âm thanh

Khối lệnh

Bài tập

Lập trình để thêm âm thanh birthday khác.

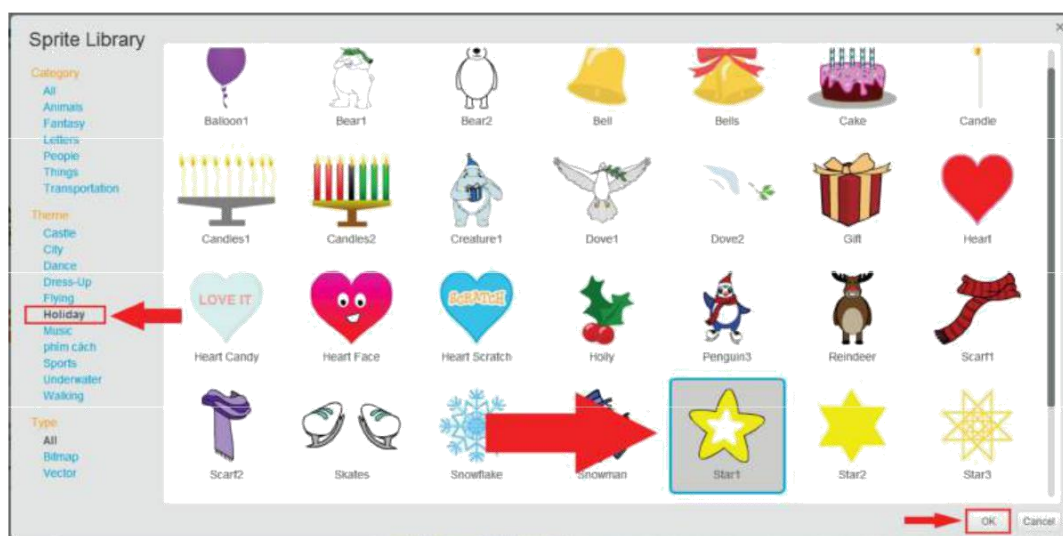
BÀI 10

LẬP TRÌNH VỚI BẢN SAO

Đối tượng cuối cùng của tấm thiệp là Ngôi sao. Bài học này sẽ hướng dẫn các bạn lập trình điều khiển ngôi sao, tạo bản sao và bản sao của ngôi sao nhấp nháy.

10.1. Thêm đối tượng Ngôi sao

Trong thư viện đã có sẵn Ngôi sao, các bạn thêm nhân vật từ thư viện, chọn mục Holiday sẽ thấy Ngôi sao cần tìm.



Hình 2.10.1. Thêm Ngôi sao từ thư viện

Sau khi thêm, các bạn có thể phóng to hoặc thu nhỏ ngôi sao để chỉnh kích thước cho phù hợp.

10.2. Tạo bản sao

Để tấm thiệp có nhiều ngôi sao, chúng ta có thể thêm nhiều nhân vật ngôi sao khác cũng giống như vậy, tuy nhiên việc đó mất rất nhiều công sức và dễ gây nhầm lẫn các nhân vật. Thay vào đó, chúng ta sẽ sử dụng khối lệnh **Tạo bản sao** tạo bản sao từ myself để tạo ra nhiều ngôi sao khác giống hệt ngôi sao ban đầu. Ở khu lập trình của Ngôi sao, các bạn chạy thử khối lệnh **Tạo bản sao**. Ta kéo ngôi sao ra sẽ thấy một ngôi sao khác xuất hiện ngay bên dưới.





Hình 2.10.2. Bản sao của Ngôi sao xuất hiện bên dưới bản chính

Ta có thể chọn nhân vật trong danh sách để tạo bản sao của nhân vật khác. Tất cả các bản sao đều giống hệt bản chính và được đặt ra sau bản chính một lớp.



Hình 2.10.3. Tạo bản sao từ nhân vật khác

10.3. Lập trình thay đổi hiệu ứng cho bản sao

Chúng ta sẽ lập trình điều khiển ngôi sao mờ đi, một giây sau sẽ hiện rõ trở lại và một giây sau nữa sẽ đổi màu. Chúng ta sẽ sử dụng khối lệnh Thay đổi hiệu ứng thay đổi hiệu ứng màu ▾ một lượng ● điều khiển việc thay đổi hiệu ứng màu và hiệu ứng mờ ảo. Với hiệu ứng mờ ảo, lượng giá trị thay đổi là số dương sẽ khiến đối tượng trở nên mờ hơn và ngược lại. Ta có thể chọn lượng giá trị thay đổi của hiệu ứng tùy ý để ngôi sao đẹp hơn.



Hình 2.10.4. Các mức độ hiệu ứng mờ ảo

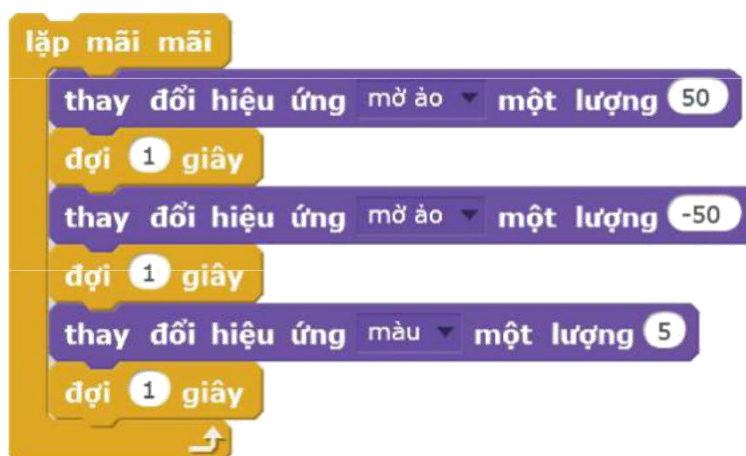


Chúng ta sẽ cho ngôi sao nhấp nháy bằng hai khối lệnh thay đổi hiệu ứng mờ ảo và đổi màu bằng một khối lệnh đổi hiệu ứng màu. Ta ghép các khối lệnh và chạy thử.



Hình 2.10.5. Thay đổi các hiệu ứng

Ghép thêm vòng lặp để việc thay đổi hiệu ứng được thực hiện liên tục.



Hình 2.10.6. Thay đổi các hiệu ứng liên tục

Tuy nhiên, đoạn khối lệnh trên chỉ giúp Ngôi sao nhấp nháy mà bản sao của nó thì không. Ghép khối lệnh **khí tôi khởi đầu là một bản sao** lên trên để bản sao của Ngôi sao nhấp nháy được như vậy.



Hình 2.10.7. Lập trình cho bản sao

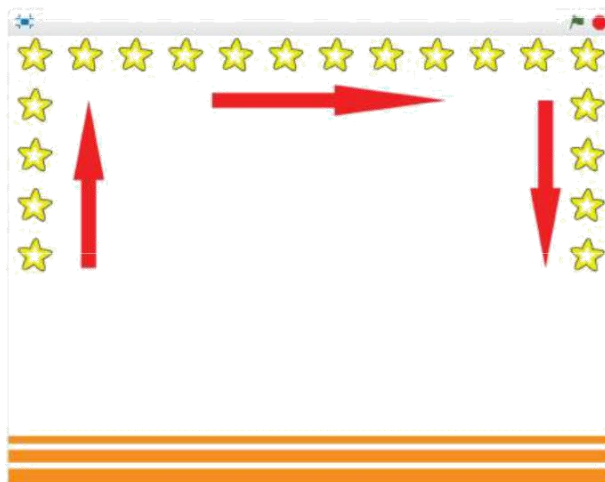
Lúc này khi các bạn tạo bản sao cho Ngôi sao, kéo Ngôi sao ra sẽ thấy bản sao của Ngôi sao thay đổi hiệu ứng liên tục.



Hình 2.10.8. Bản sao thay đổi hiệu ứng

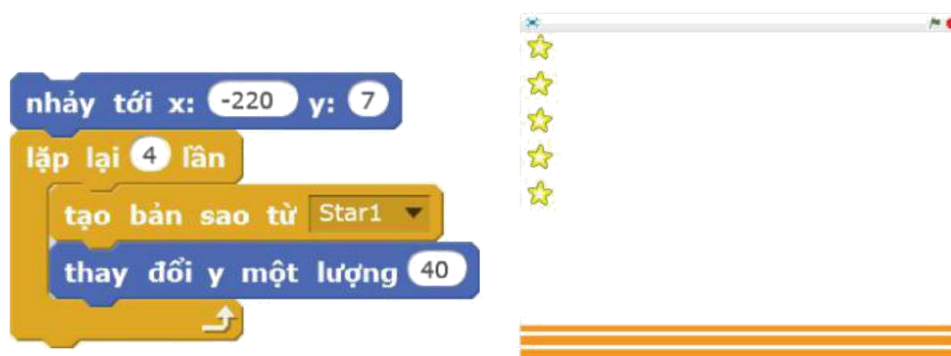
10.4. Sắp xếp các bản sao của Ngôi sao

Trong tám thiệp, các ngôi sao được sắp xếp thành hai cột và một hàng giống như ba cạnh của hình chữ nhật. Ta lập trình cho ngôi sao xếp lần lượt từng cạnh một bằng cách lặp lại công việc tạo bản sao và di chuyển sang vị trí tiếp theo.



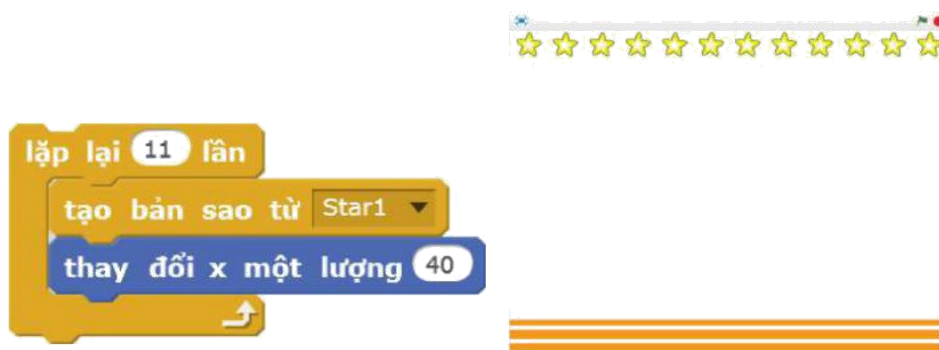
Hình 2.10.9. Sắp xếp các bản sao

Với cột đầu tiên, Ngôi sao được lập trình di chuyển từ dưới lên trên, tức là giá trị y của Ngôi sao được tăng (điều chỉnh số bước di chuyển và số lần lặp tùy theo kích thước và vị trí của ngôi sao).



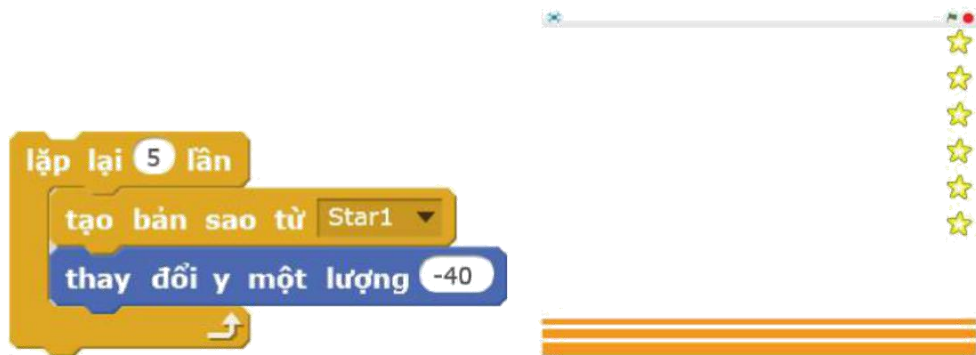
Hình 2.10.10. Cột sao thứ nhất

Hàng tiếp theo, Ngôi sao di chuyển từ trái sang phải, tức là giá trị x của ngôi sao được tăng.



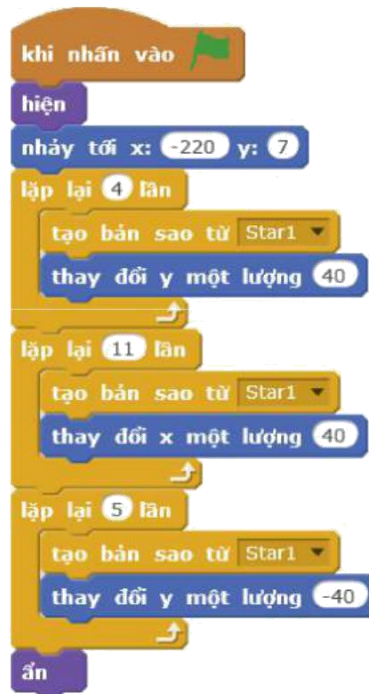
Hình 2.10.11. Hàng sao

Cột cuối cùng, Ngôi sao di chuyển từ trên xuống dưới, tức là giảm giá trị y của Ngôi sao.



Hình 2.10.12. Cột sao thứ 2

Chỉ bản sao mới có hiệu ứng nên sau khi tạo đủ bản sao rồi, các bạn ẩn ngôi sao đi bằng khối lệnh **Ẩn** trong nhóm lệnh **Ngoại hình**. Tuy nhiên để tránh việc khi chạy lại chương trình lần sau Ngôi sao vẫn bị ẩn thì ta thêm khối lệnh **Hiện** vào đầu đoạn khối lệnh. Cuối cùng là khối lệnh kích hoạt và đặt vị trí bắt đầu cho Ngôi sao.



Hình 2.10.13. Đoạn khối lệnh hoàn chỉnh

Chương thiết kế **Thiệp sinh nhật** đến đây là kết thúc, chúng ta đã được trang bị những kiến thức cơ bản để có thể tự thiết kế một tấm thiệp cho riêng mình.



Tóm tắt lí thuyết và bài tập thực hành

Lập trình với bản sao

Tạo bản sao và hiệu ứng đi kèm

tạo bản sao từ **myself**

thay đổi hiệu ứng màu **một lượng**

daylaptrinh.net

Đối tượng

Bản sao thay đổi hiệu ứng

Khối lệnh

```

lặp lại 5 lần
  tạo bản sao từ Star1
  thay đổi y một lượng -40
        
```

Bài tập

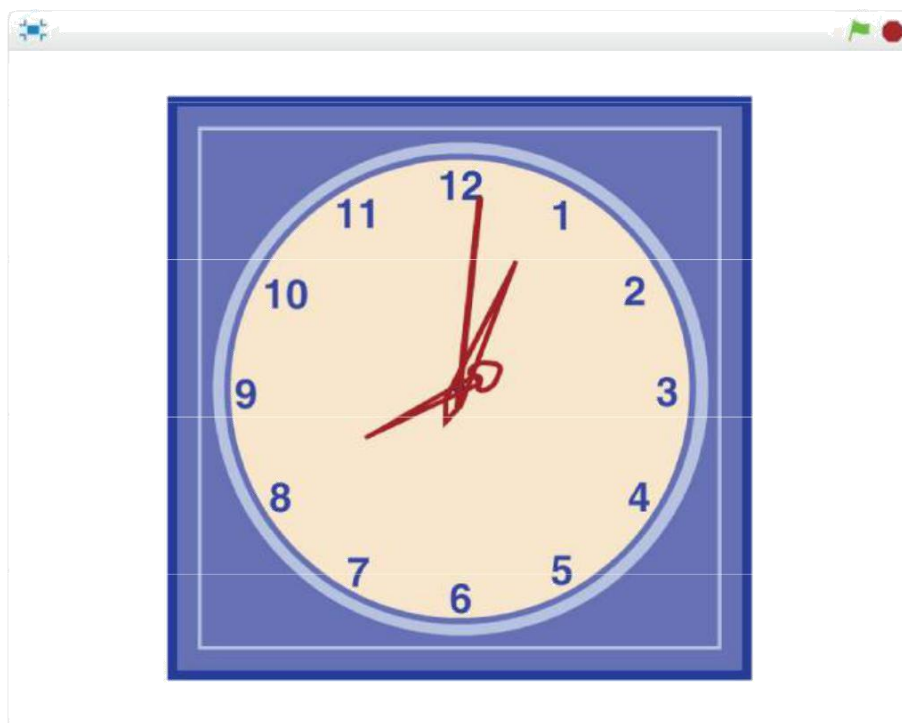
- Lập trình để bản sao của Ngôi sao nhấp nháy và đổi màu nhanh hơn.
- Lập trình để bản sao của Ngôi sao thay đổi được độ sáng.



Đồng hồ

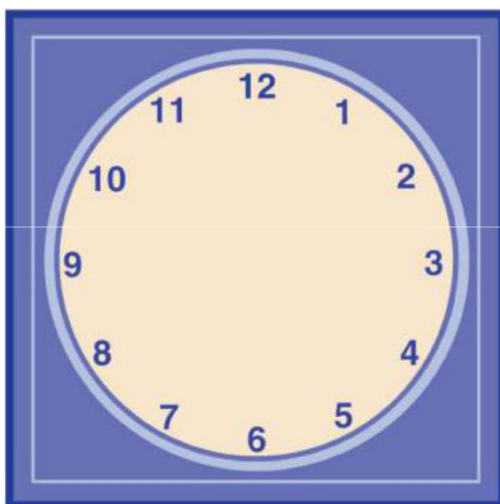
Trong chương này, các bạn sẽ được hướng dẫn để tạo ra một chiếc đồng hồ kim. Sau đó, các bạn sẽ học được cách thiết kế các thành phần của đồng hồ và lập trình cho các kim để đồng hồ hoạt động với thời gian chính xác như thời gian được cài đặt trên máy tính.

Tổng quan chương trình



Chiếc đồng hồ gồm bốn đối tượng thành phần: Mặt đồng hồ, kim giờ, kim phút và kim giây.

1. Mặt đồng hồ



Hình dáng, màu sắc và các số trên đồng hồ được biểu diễn ở đối tượng Mặt đồng hồ.

2. Kim giờ



Đối tượng Kim giờ mô phỏng chuyển động của kim giờ.

3. Kim phút



Đối tượng Kim phút mô phỏng chuyển động của kim phút.

4. Kim giây



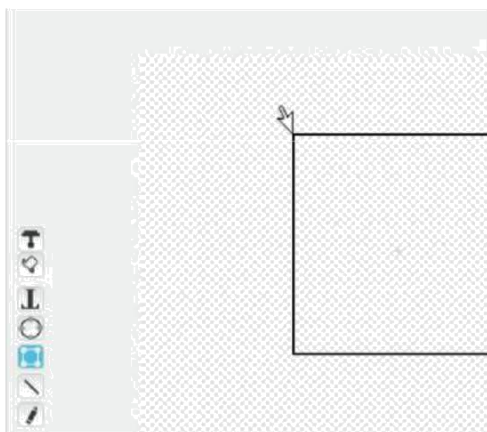
Đối tượng Kim giây mô phỏng chuyển động của kim giây.



Hình 3.1.2. Chỉnh tâm hình vuông bằng nút **Điều chỉnh tâm**

Chỉnh lại tâm hình vuông trùng với điểm chấm tròn ở chính giữa. Tuy nhiên, khi nháy chuột vào nút **Điều chỉnh tâm**, điểm chấm tròn bao quanh hình vuông biến mất.

Hình 3.1.1. Vẽ hình vuông



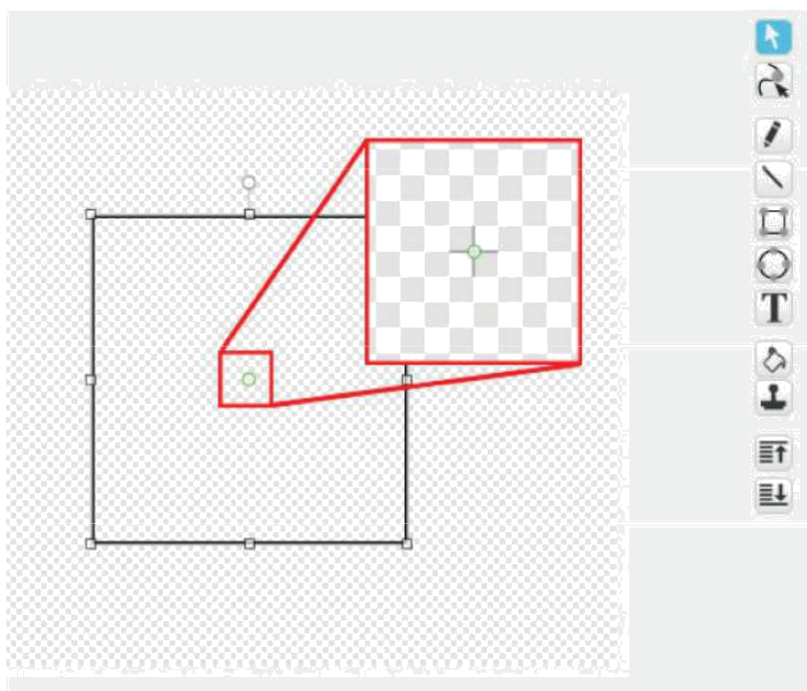
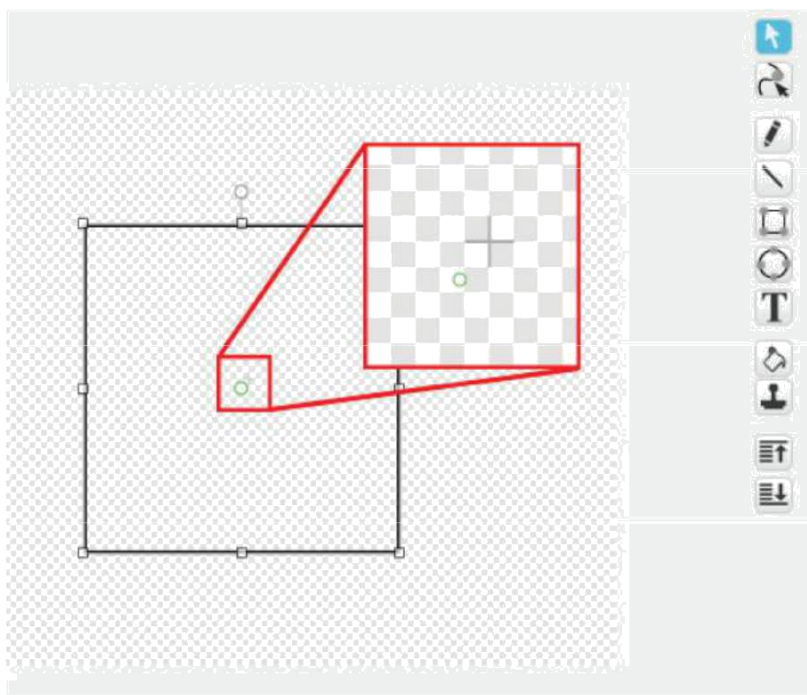
Mặt đồng hồ được tạo bởi hai hình vuông và hai đường tròn bên trong. Trước tiên, chúng ta cần vẽ đường viền hình vuông ngoài cùng bằng cách chọn công cụ vẽ **Hình chữ nhật**, kéo thả chuột đồng thời nhấn giữ phím **Shift** để vẽ được hình vuông.

1.1. Vẽ mặt đồng hồ

VỀ MẶT ĐỒNG HỒ

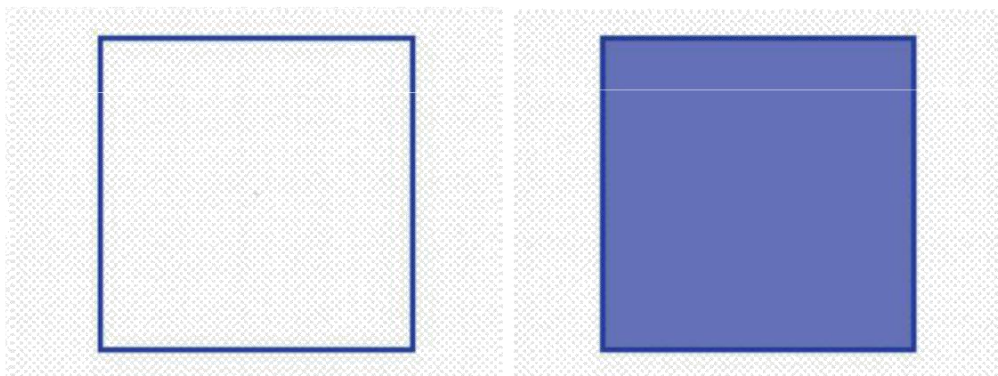
BÀI 1

Di chuyển hình vuông trên trang vẽ để điểm chính giữa hình vuông trùng với tâm trang vẽ. Các bạn có thể sử dụng chức năng phóng to  để điều chỉnh dễ dàng hơn.



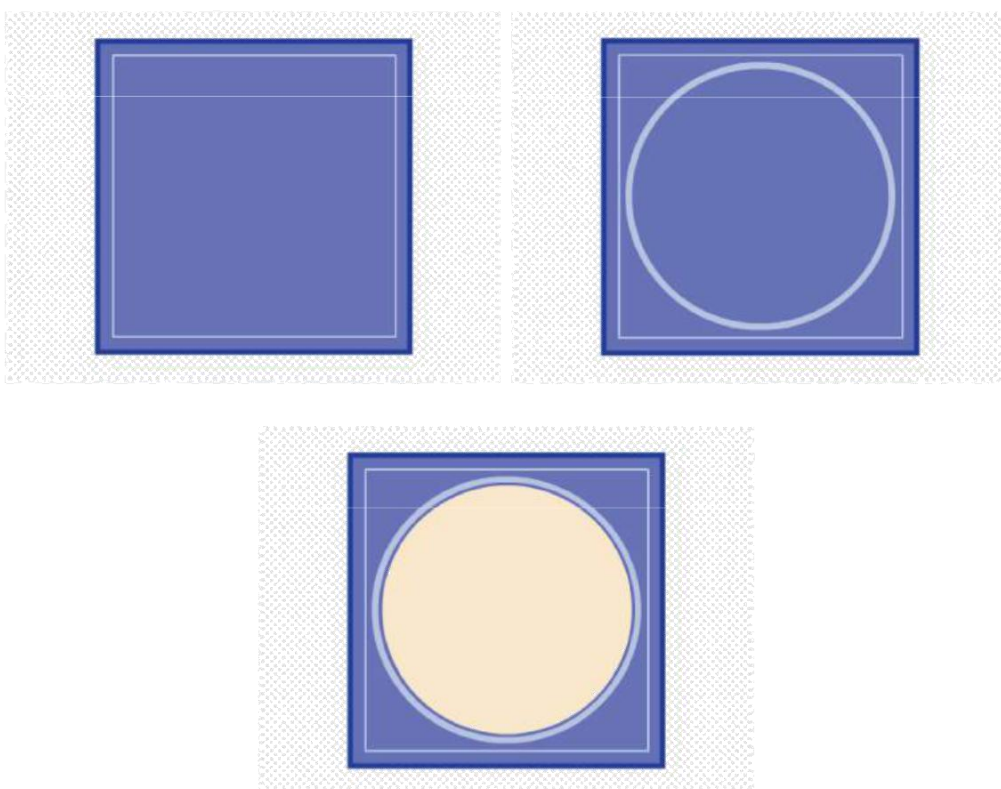
Hình 3.1.3. Chỉnh tâm hình vuông bằng tâm trang vẽ

Tô màu bên trong hình vuông bằng công cụ , chỉnh màu và kích thước nét vẽ đường viền tùy ý.



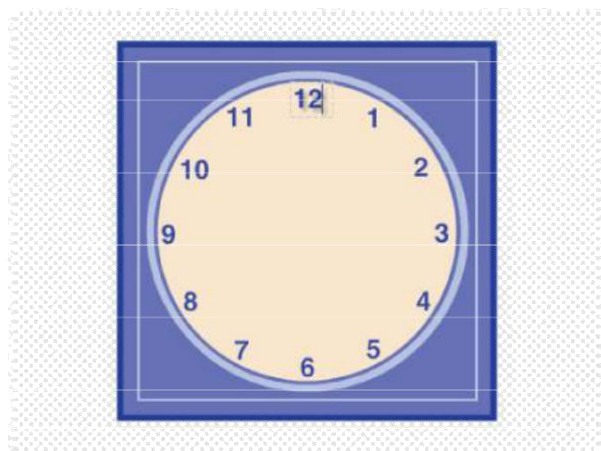
Hình 3.1.4. Chỉnh kích thước, màu đường viền và đổ màu

Vẽ đường viền hình vuông thứ hai và thêm hai đường tròn. Vẽ hình tròn bằng công cụ vẽ hình elip, khi vẽ nhấn phím **Shift** để có được hình tròn. Thao tác chỉnh tâm và chỉnh màu được thực hiện tương tự.



Hình 3.1.5. Thêm một đường viền hình vuông và hai đường tròn

Thêm số cho đồng hồ. Dùng công cụ **T** để thêm các số từ 1 đến 12 và điều chỉnh vị trí của số cho phù hợp.



Hình 3.1.6. Thêm số cho đồng hồ



Tóm tắt lí thuyết và bài tập thực hành

Vẽ mặt đồng hồ

Cách vẽ mặt đồng hồ

daylaptrinh.net

Các bước thực hiện

Công cụ

Vẽ hình chữ nhật

Vẽ hình elip

Thêm văn bản

Bài tập

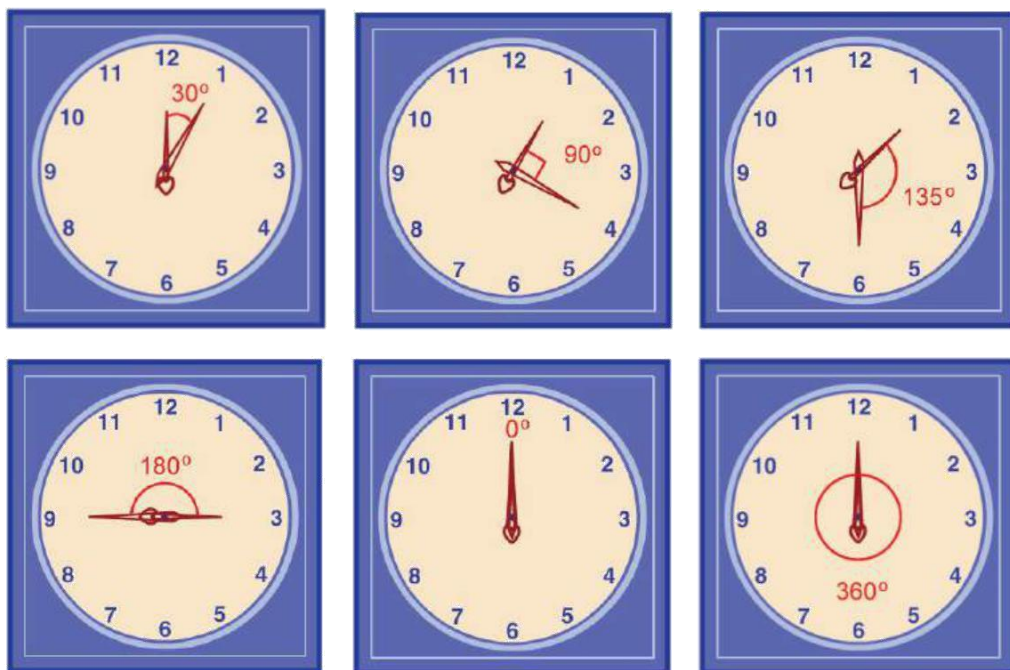
- Vẽ thêm tâm cho đồng hồ sao cho thật chính xác.
- Tự sáng tạo một chiếc đồng hồ có hình dáng và màu sắc khác.

BÀI 2 GIỚI THIỆU VỀ GÓC VÀ HƯỚNG CỦA ĐỐI TƯỢNG

2.1. Giới thiệu về góc

Cứ hai kim bất kì trong đồng hồ đều sẽ tạo thành một góc. Đơn vị để đo độ lớn của góc là độ, kí hiệu $^{\circ}$. Ví dụ 30° (30 độ). Góc cũng dùng để mô tả chuyển động xoay. Góc xoay càng rộng thì giá trị góc càng lớn. Xoay hết một vòng tròn thì góc bằng 360° .

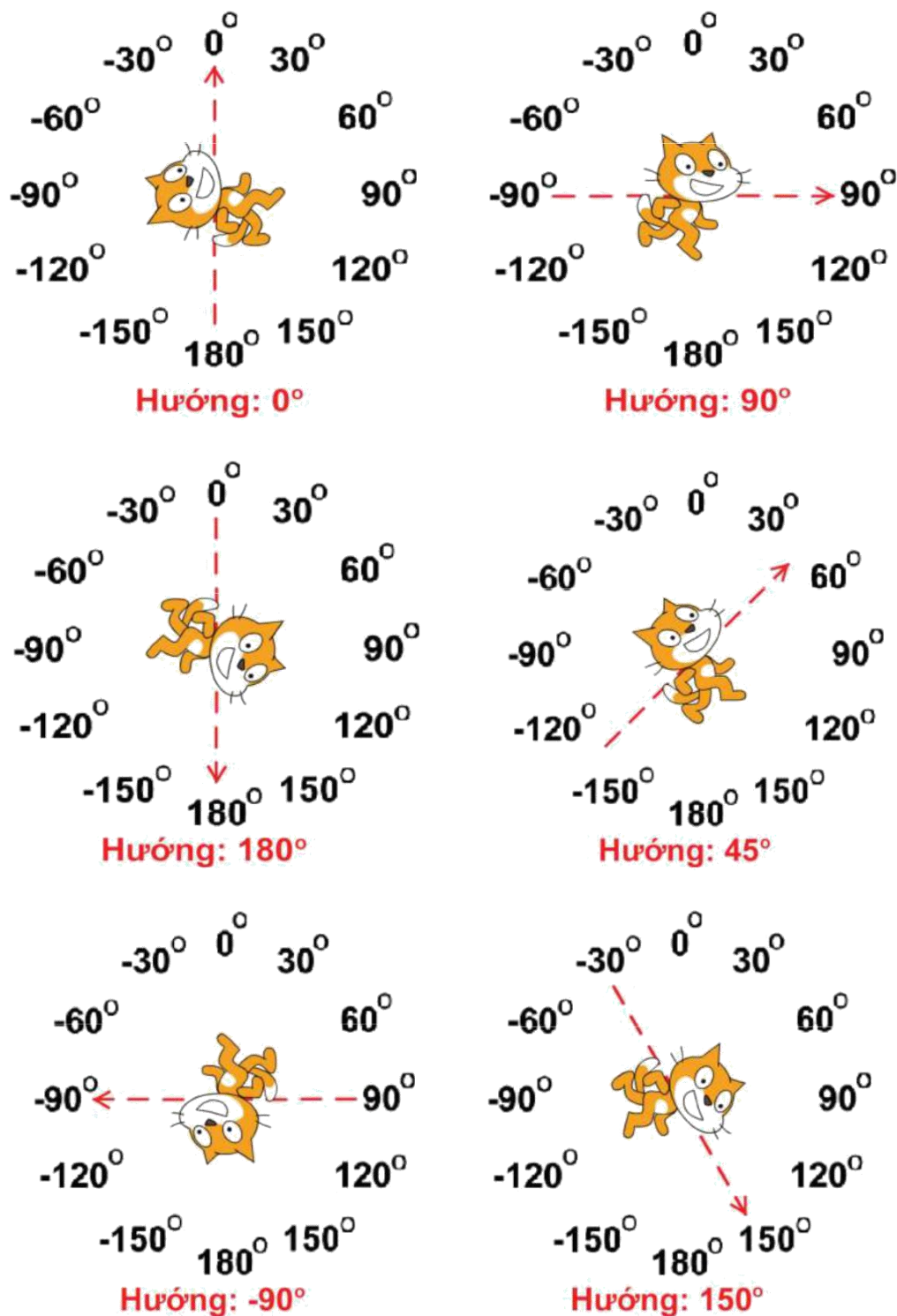
Hình 3.2.1 minh họa các giá trị góc được tạo bởi kim giờ và kim phút của đồng hồ. Lưu ý có thể có một số trường hợp không đúng với thực tế.



Hình 3.2.1. Các góc được tạo bởi kim giờ và kim phút

2.2. Hướng của đối tượng

Mỗi đối tượng vật đều có một hướng nhất định và có giá trị ngầm định là hướng sang phải (90°). Hướng và giá trị hướng của đối tượng được mô tả như trên Hình 3.2.2.



Hình 3.2.2. Một số ví dụ về hướng của đối tượng

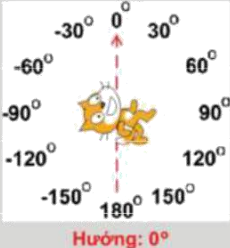


Tóm tắt lí thuyết và bài tập thực hành

daylaptrinh.net

Góc và Hướng

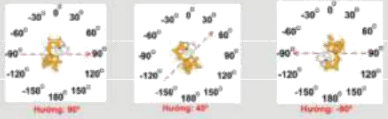
Giới thiệu về góc và hướng của đối tượng

Góc



Hướng



Bài tập

- Xác định góc của kim giờ và kim phút ở thời điểm 2 giờ, 10 giờ, 8 giờ.
- Thực hành thay đổi hướng của nhân vật bằng các khối lệnh xoay.

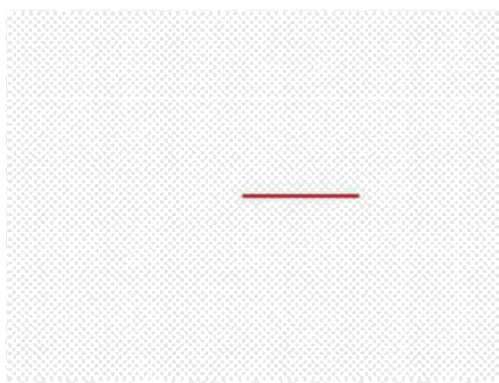
BÀI 3

VẼ KIM ĐỒNG HỒ

Sau khi vẽ được mặt đồng hồ, các bạn cần thiết kế các kim đồng hồ và chỉnh lại vị trí các số cho chính xác.

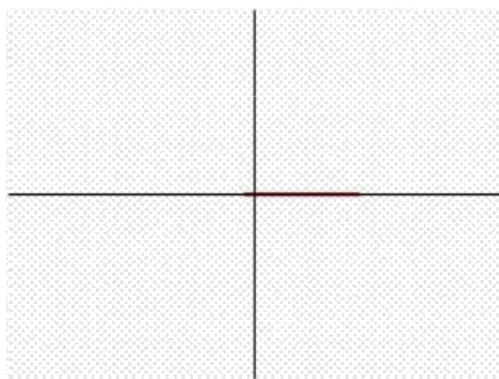
3.1. Kim giây

Hình dáng kim giây rất đơn giản, chỉ là một đường thẳng. Các bạn dùng công cụ , chỉnh màu và kích thước nét vẽ rồi sau đó tiến hành vẽ. Khi vẽ nhấn giữ phím **Shift**.



Hình 3.3.1. Vẽ kim giây

Trên mặt đồng hồ, điểm quay của các kim không nằm ở chính giữa kim mà sẽ lệch về một bên, do đó các bạn cần chỉnh tâm của kim cũng lệch tương ứng như vậy.



Hình 3.3.2. Chỉnh tâm của kim

3.2. Kim phút

Cách vẽ kim phút phức tạp hơn một chút. Sau khi tạo đối tượng mới, các bạn vẽ đường viền hình chữ nhật, sau đó tạo hình bằng công cụ chỉnh hình.

Khi ở chế độ chỉnh hình có các chấm tròn xuất hiện trên đối tượng, nhấp chuột vào một điểm để bỏ điểm đó, nhấn vào một cạnh để tạo thêm một điểm.

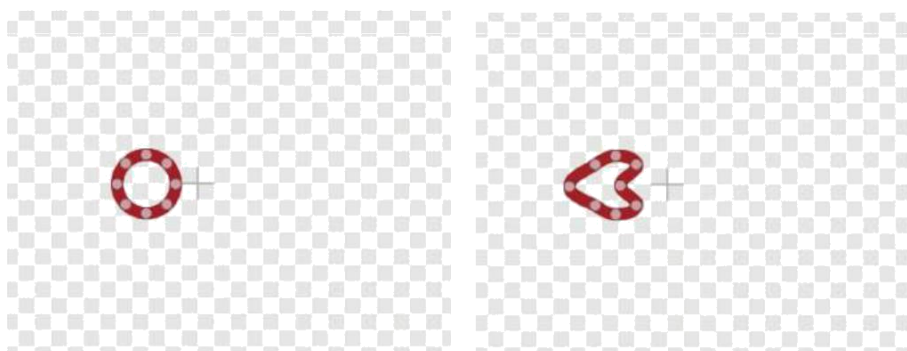




Hình 3.3.3. Tạo hình kim phút

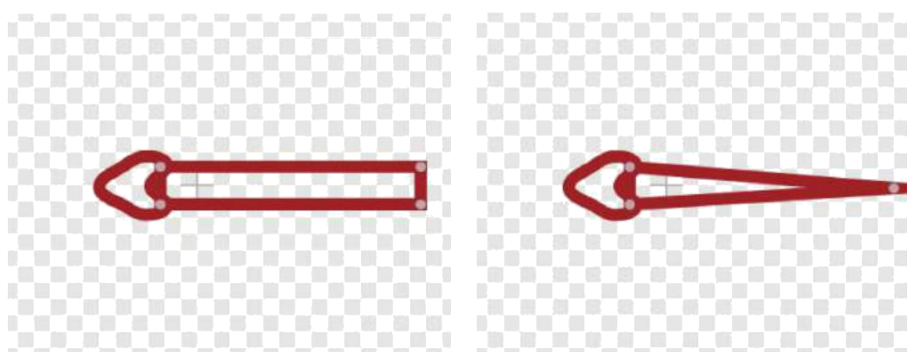
3.3. Kim giờ

Đầu tiên vẽ một đường tròn, chỉnh hình để thành hình trái tim.



Hình 3.3.4. Tạo hình đầu kim giờ

Thêm một hình chữ nhật và chỉnh lại hình để tạo thành đầu kim. Sau khi vẽ xong, đổi lại tên cho các đối tượng.

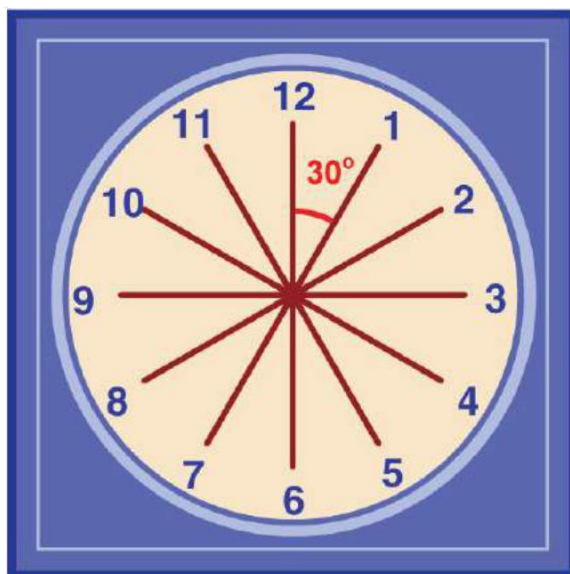


Hình 3.3.5. Kim giờ

3.4. Chỉnh lại vị trí các số

Ở bài trước, các bạn đã thêm số cho mặt đồng hồ, tuy nhiên vị trí mới chỉ ước lượng, chưa chính xác.

Nổi 12 số với tâm sẽ được 12 góc, mỗi góc bằng 30° . Các bạn sẽ lập trình để kim giây xoay 12 lần, mỗi lần quay 30° và tạo bản sao. Sau đó sắp xếp lại vị trí các số trên mặt đồng hồ.



Hình 3.3.6. Vị trí các số

Đầu tiên lập trình để kim giây nhảy tới tâm của đồng hồ. Ở khu lập trình của kim giây, chúng ta sử dụng khối lệnh **nhảy tới** con trỏ chuột và chọn đối tượng đồng hồ.



Hình 3.3.7. Kim giây nhảy tới tâm của đồng hồ

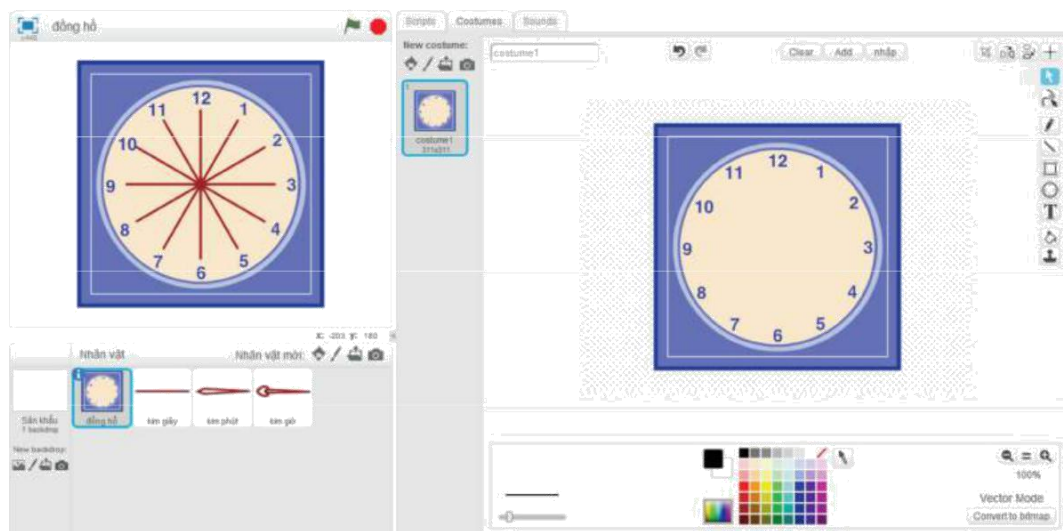
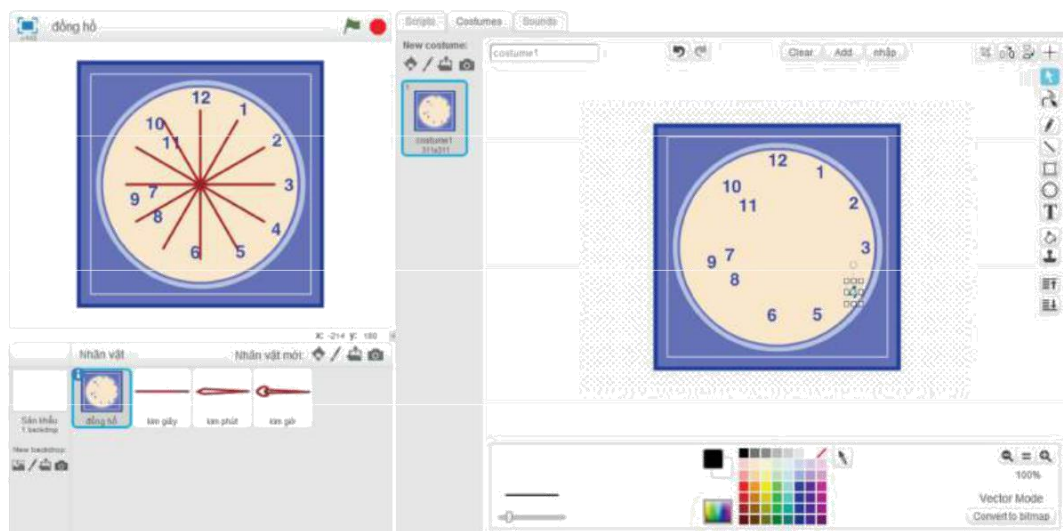
Tiếp theo, lập trình cho kim giây xoay và tạo bản sao.



Hình 3.3.8. Kim giây tạo bản sao

Cuối cùng, vào trang **Costumes** của đối tượng **Đồng hồ** để và sắp xếp lại vị trí các số.



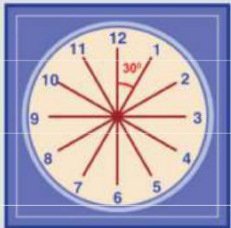


Hình 3.3.9. Sắp xếp vị trí các số của đồng hồ



Vẽ kim đồng hồ

Vẽ kim đồng hồ và chỉnh lại vị trí các số trên mặt đồng hồ



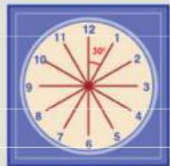
daylaptrinh.net

Vẽ kim



Kim giây Kim phút Kim giờ

Chỉnh vị trí các số



Bài tập

1. Lập trình để khi nhấp chuột vào  thì mặt đồng hồ sẽ nhảy tới tọa độ (0, 0).
2. Lập trình để khi nhấp chuột vào  thì ba kim sẽ nhảy tới tâm của đồng hồ.



BÀI 4

LẬP TRÌNH KIM GIÂY

4.1. Hướng của đối tượng

Mặc định hướng của nhân vật là hướng 90° , tức là hướng sang phải. Các bạn có thể điều khiển để đối tượng hướng về một phía nhất định bằng khối lệnh **xoay về phía 90**. Chúng ta có thể thay đổi giá trị góc xoay tùy ý. Giá trị góc này chính là hướng của nhân vật đã được nhắc đến ở Bài 2 và được minh họa ở Hình 3.2.2.

4.2. Lập trình kim giây

Khối lệnh **hiện tại date** chứa đầy đủ thông tin về ngày, tháng, năm, giờ, phút, giây của đồng hồ trên máy tính. Chạy thử khối lệnh và ta sẽ thấy được giá trị hiện tại của thời gian tương ứng.



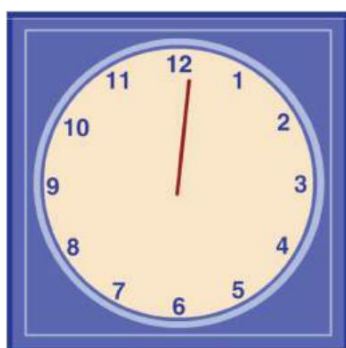
Hình 3.4.1. Khối lệnh thời gian với năm hiện tại và giờ hiện tại

Sau khi đã biết được giây hiện tại bằng cách chọn **Second** trong khối lệnh **hiện tại second**, chúng ta cần tính toán xem hướng mà kim giây phải xoay ứng với mỗi giây.

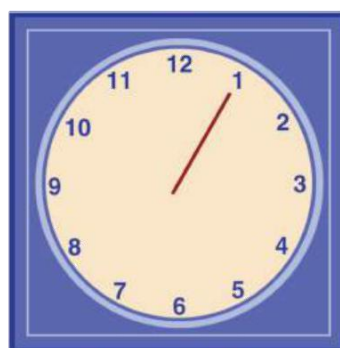
Để quay hết một vòng tròn (360°) kim giây phải quay 60 lần, vậy mỗi lần kim giây sẽ quay một góc có giá trị bằng $360^\circ/60 = 6^\circ$.

Giây đầu tiên, kim quay về hướng góc 6° , giây tiếp theo là 12° và cứ như vậy.

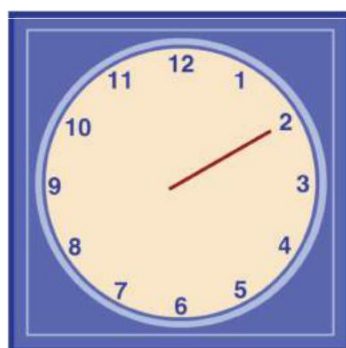




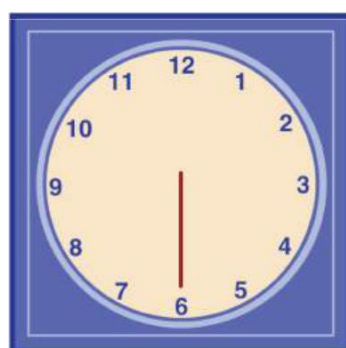
Giây thứ 1
Hướng: 6°



Giây thứ 5
Hướng: 30°



Giây thứ 10
Hướng: 60°



Giây thứ 30
Hướng: 180°

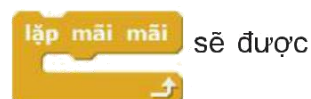
Hình 3.4.2. Hướng của kim giây

Vậy giá trị góc của hướng sẽ bằng giây hiện tại nhân với 6, ghép lại bằng cách kéo thả khối lệnh **hiện tại** **second** lên khối lệnh ***** khi nổi viền trắng thì thả chuột, các bạn sẽ có được giá trị góc cần phải xoay **hiện tại** **second** ***** **6**.

Lưu ý: ghép đầu bên trái của khối lệnh hiện tại vào đầu bên trái của ô bên trong khối lệnh tính nhân.

Ghép giá trị hướng vừa tính được vào khối lệnh xoay **xoay về phía** **90** để điều khiển kim giây xoay về đúng góc.

Việc xoay này phải luôn được thực hiện, vì vậy khối lệnh sử dụng.



Các bạn nhớ ghép vào dưới khối lệnh **khi nhấn vào** để kim có thể hoạt động khi chúng ta nhấp lệnh.





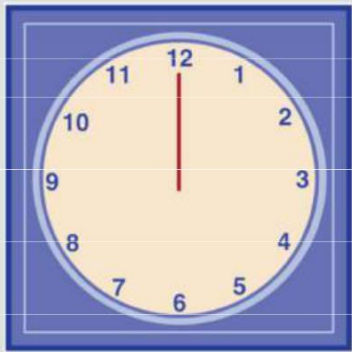
Hình 3.4.3. Đoạn khối lệnh chuyển động của Kim giây



Tóm tắt lí thuyết và bài tập thực hành

Lập trình cho kim giây

Lập trình cho kim giây chuyển động



daylaptrinh.net

Lấy thời gian trên máy tính

20

hiện tại second

Khối lệnh chuyển động



Bài tập

Lập trình để kim giây luôn luôn nhảy tới mặt đồng hồ và luôn được đặt ra trước mặt đồng hồ.

BÀI 5

LẬP TRÌNH KIM PHÚT

Để quay hết một vòng tròn (360°) kim phút phải quay 60 lần, vậy mỗi lần kim phút sẽ quay một góc có giá trị bằng $360^\circ/60 = 6^\circ$ và giống với kim giây. Vậy chúng ta chỉ cần lập trình giống với kim giây, nhưng thay vì lấy giá trị giây **hiện tại** `second` các bạn sẽ lấy giá trị phút hiện tại **hiện tại** `minute`.



Đoạn khối lệnh chuyển động của kim phút



Tóm tắt lí thuyết và bài tập thực hành

Lập trình cho kim phút

Lập trình cho kim phút chuyển động

daylaptrinh.net

Lấy thời gian từ máy tính

20

hiện tại minute

Khối lệnh chuyển động

```

when clicked
  loop forever
    turn right (6) degrees
          
```

Bài tập

Lập trình để kim phút luôn luôn nhảy tới mặt đồng hồ và luôn được đặt ra trước mặt đồng hồ.

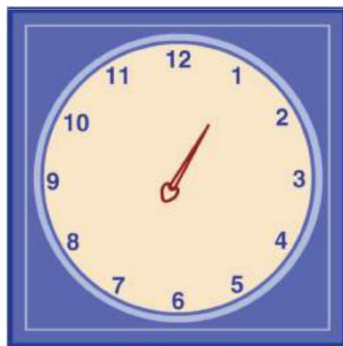


BÀI 6

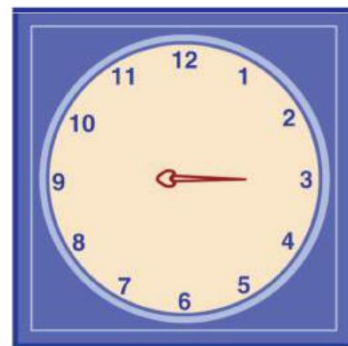
LẬP TRÌNH KIM GIỜ

Kim giờ quay hết một vòng 360° ứng với 12 giờ, vậy mỗi giờ kim giờ phải xoay thêm một góc bằng $360^\circ/12 = 30^\circ$.

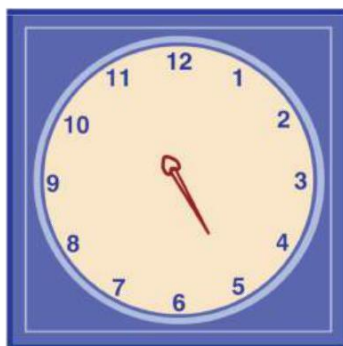
Vậy thì giá trị góc của hướng ứng với giờ hiện tại sẽ bằng giờ hiện tại nhân với 30 là * .



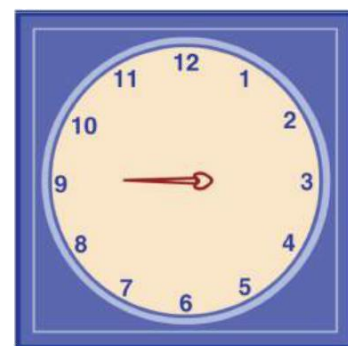
1 giờ
Hướng: 30°



3 giờ
Hướng: 90°



5 giờ
Hướng: 150°

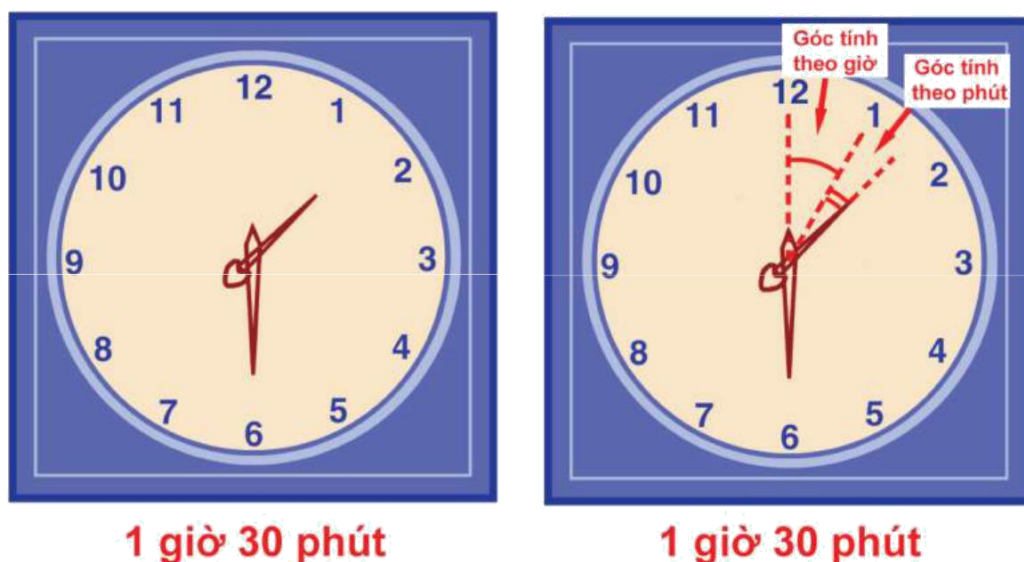


9 giờ
Hướng: $270^\circ (-90^\circ)$

Hình 3.6.1. Hướng của kim giờ

Tuy nhiên, ở chiếc đồng hồ thật thì khi chuyển giữa các giờ, kim giờ sẽ xoay từng chút một chứ không xoay đột ngột mỗi lần 30° , tức là giá trị góc của hướng sẽ bằng góc tính theo giờ hiện tại cộng với góc tính theo phút hiện tại.





Hình 3.6.2. Hướng của kim giờ tính thêm góc theo phút

Khi kim phút quay một vòng đủ 60 phút (hết một giờ) thì kim giờ sẽ xoay thêm một góc 30° , vậy mỗi phút kim giờ sẽ xoay một góc 30° chia 60 bằng 0.5° .

Vậy góc tính theo phút sẽ bằng phút hiện tại nhân với 0.5 là

* và tổng hai góc đó sẽ là hướng của kim giờ.



Hình 3.6.3. Đoạn khối lệnh chuyển động của kim giờ

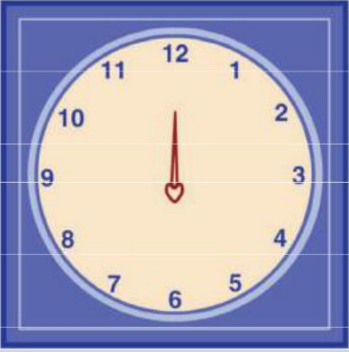
Chiếc đồng hồ đến đây đã được hoàn thành, các bạn có thể sáng tạo thêm bằng cách thêm quả lắc hoặc thêm chức năng hẹn giờ cho đồng hồ. Chúc các bạn thành công !



Tóm tắt lí thuyết và bài tập thực hành

Lập trình cho kim giờ

Lập trình cho kim giờ chuyển động



daylaptrinh.net

Lấy thời gian từ máy tính

15

hiện tại: hour



Khởi lệnh chuyển động

khí nhân vào

lấy một mẫu

xoay về phía

Hiện tại: hour * 30 + Hiện tại: minute * 0.5

Bài tập

Lập trình để kim giờ luôn luôn nhảy tới mặt đồng hồ và luôn được đặt ra trước mặt đồng hồ.



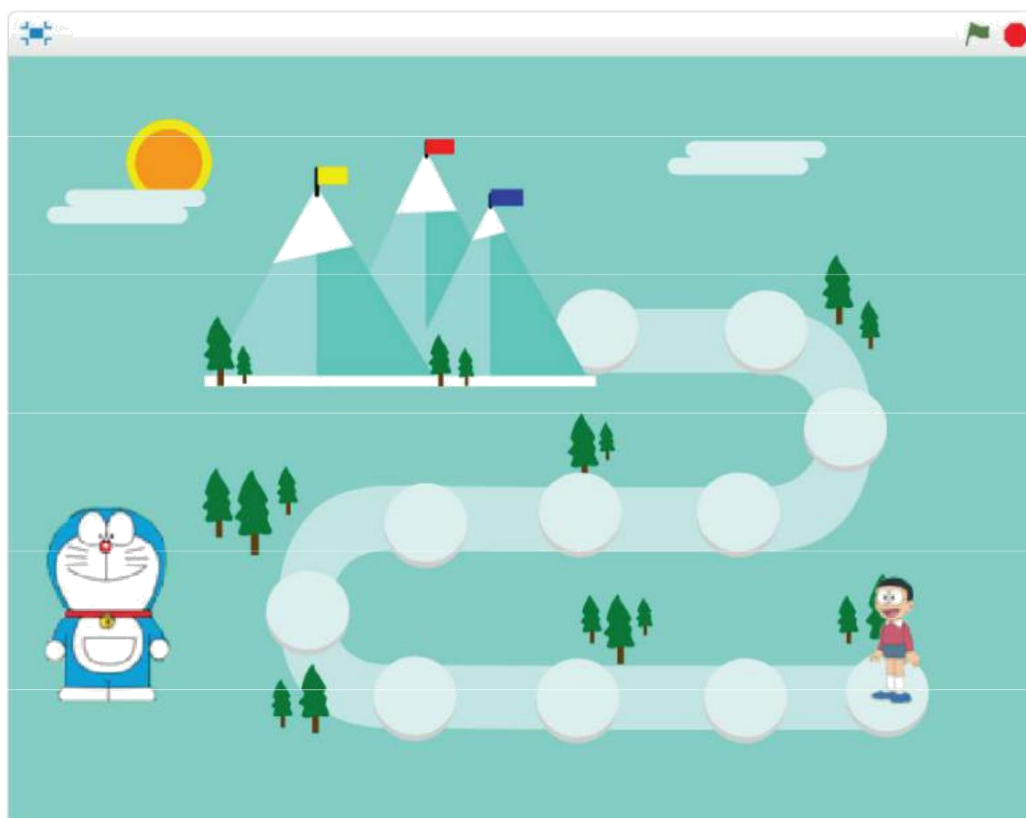
CHƯƠNG

IV

Đường lên đỉnh Olympia

Trong chương này, các bạn sẽ được hướng dẫn để tạo ra một trò chơi rất hấp dẫn. Trong trò chơi này, các bạn sẽ đóng vai Nobita trong bộ phim hoạt hình Doraemon trả lời các câu hỏi là các phép tính để tiến lên đỉnh Olympia.

Tổng quan chương trình



Trong trò chơi, người chơi sẽ phải liên tục trả lời các câu hỏi toán học do nhân vật Doraemon đưa ra. Với mỗi câu trả lời đúng, nhân vật Nobita sẽ được tiến lên một bước. Để giành được chiến thắng, người chơi sẽ phải đưa Nobita lên đến đỉnh núi.

Chương trình gồm có các nhân vật, đối tượng:

1. Doraemon



Nhân vật Doraemon sẽ liên tục đưa ra các câu hỏi toán học, người chơi nhập câu trả lời từ bàn phím.

2. Nobita



Mỗi khi người chơi trả lời đúng một câu hỏi của Doraemon, nhân vật Nobita sẽ được tiến lên một bước. Nếu Nobita tiến được đến đỉnh núi thì người chơi sẽ giành chiến thắng.

Một số chức năng của chương trình như thêm âm thanh, thay đổi hình dạng Nobita khi đến đích... sẽ không được hướng dẫn mà sẽ để trong phần Bài tập thực hành ở cuối mỗi bài. Các bạn cần làm bài tập đầy đủ để hoàn thiện chương trình.

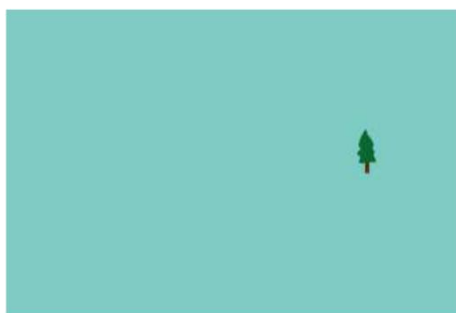
BÀI 1

VẼ ĐƯỜNG LEO NÚI

Ở những bài đầu tiên, các bạn sẽ được hướng dẫn để tạo bản đồ trò chơi.

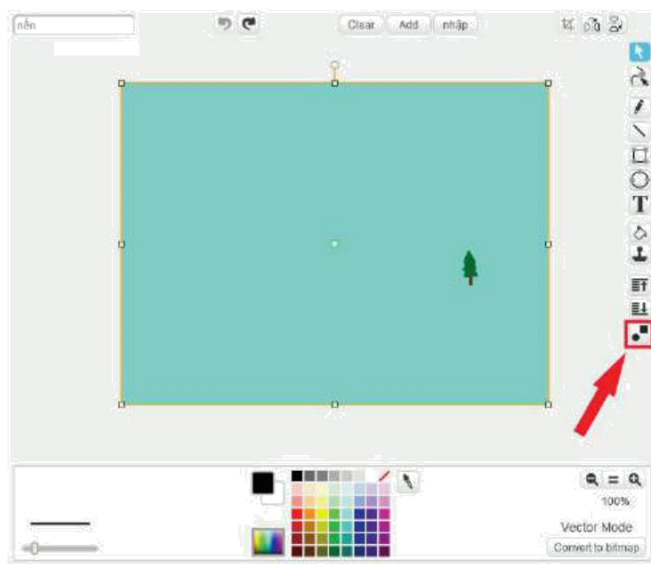
1.1. Vẽ đường leo núi

Trong phần tài liệu của bài học trên trang chủ www.daylaptrinh.net đã có sẵn hình nền là hình bản đồ đơn giản. Sau khi tải về và thêm vào chương trình, các bạn sẽ phải vẽ thêm đường leo núi vào bản đồ.



Hình 4.1.1. Bản đồ đơn giản

Trong khung nền sân khấu, nhấn vào  để tải hình nền vào, chuyển đến khu thiết kế phông nền (Backdrop) bản đồ đã có sẵn một cái cây. Sử dụng công cụ **Tách nhóm** , tách cái cây ra khỏi bản đồ, từ đó có thể thao tác với cái cây đó như một đối tượng độc lập.



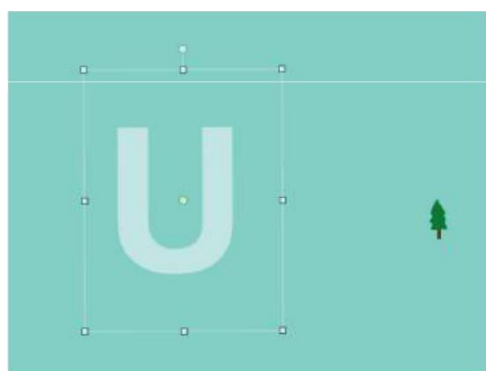
Hình 4.1.2. Tách cây khỏi bản đồ



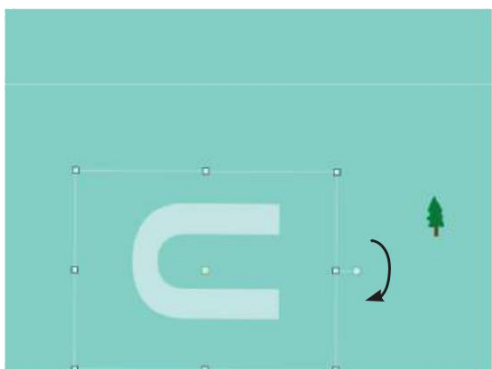
Đường leo núi gồm 2 chữ U quay mặt vào nhau và lệch nhau một đoạn. Ta sử dụng công cụ **T** để viết một chữ U, sau đó phóng to và xoay ngang là được một nửa đường. Chữ U còn lại các bạn tạo bản sao từ chữ U đầu tiên rồi xoay ngược lại và sắp xếp lại. Thêm hình chữ nhật và hoàn thiện con đường theo hình sau:



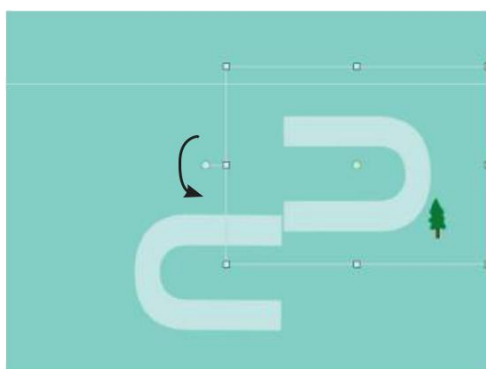
B1. Viết chữ



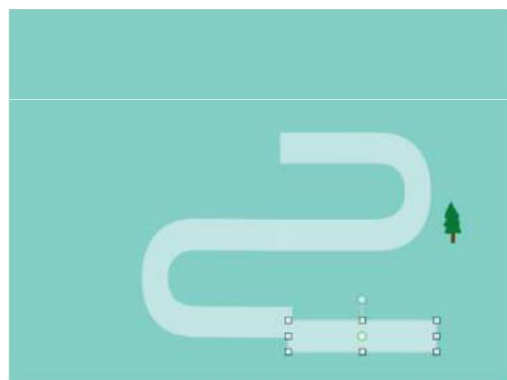
B2. Phóng to



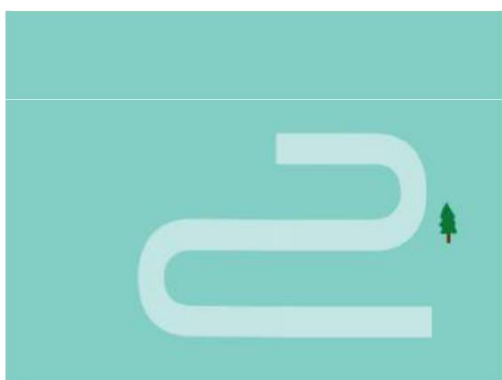
B3. Xoay ngang



B4. Nhân bản và xoay ngược lại



B5. Thêm hình chữ nhật

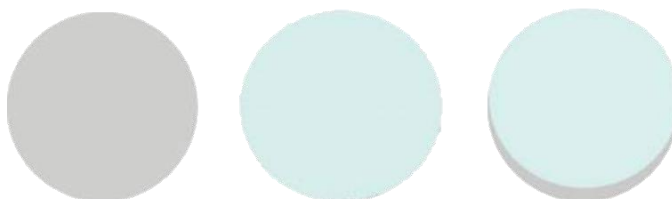


B6. Sắp xếp lại để có con đường

Hình 4.1.3. Các bước vẽ con đường

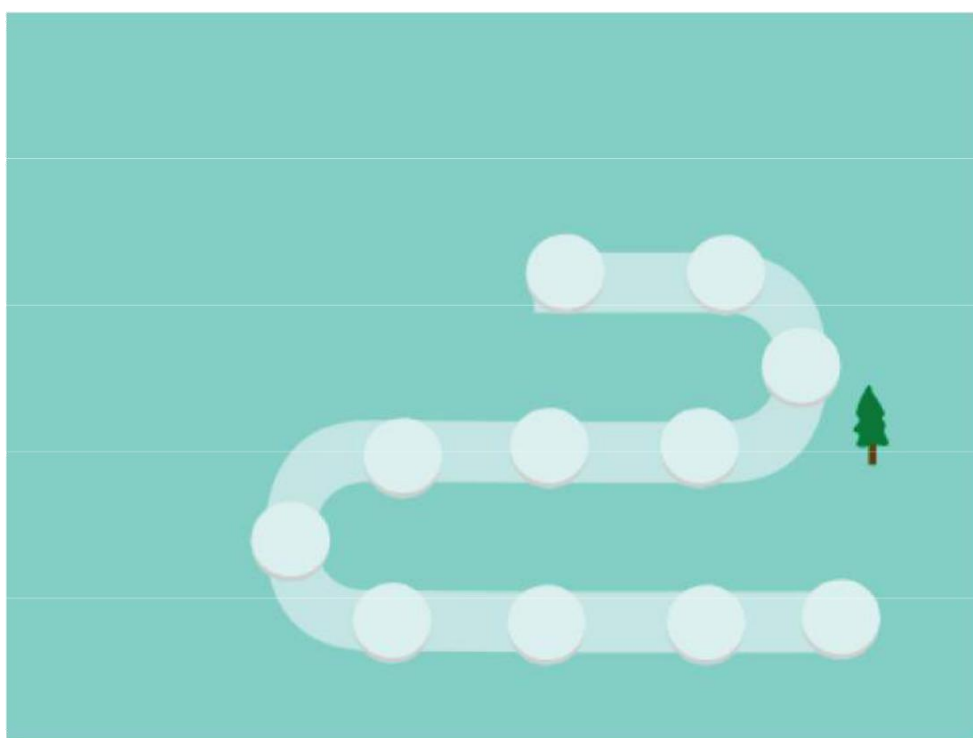
1.2. Vẽ bậc

Ta sẽ xây các bậc trên đường đi của nhân vật Nobita bằng cách sử dụng công cụ vẽ . Bậc gồm một hình tròn màu sáng và một hình tròn màu xám. Ta đặt hình tròn màu sáng lên trên hình tròn xám và sắp xếp hơi lệch nhau để hai hình tròn tạo thành một bậc có chiều cao.



Hình 4.1.4. Vẽ bậc

Sử dụng công cụ  để tạo bản sao một bậc thành nhiều bậc và sắp xếp lại vị trí như hình sau:



Hình 4.1.6. Tạo bản sao và sắp xếp bậc



Tóm tắt lí thuyết và bài tập thực hành

Vẽ đường leo núi

Vẽ đường leo núi



daylaptrinh.net

Vẽ đường



Vẽ bậc



Bài tập

1. Tạo bản sao của cây, sắp xếp lại vị trí và thay đổi kích thước các cây.
2. Thêm bậc cho bản đồ.
3. Vẽ mặt trời bằng công cụ vẽ hình tròn.

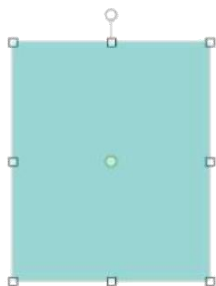


BÀI 2

VẼ NÚI VÀ MÂY

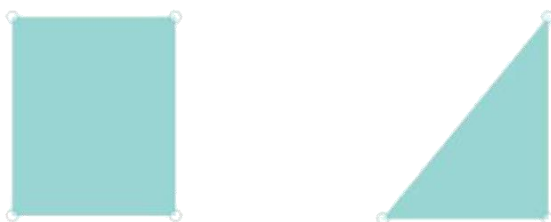
2.1. Vẽ núi

Đầu tiên vẽ một hình chữ nhật, ta có thể phóng to trang vẽ để có thể vẽ dễ dàng hơn.



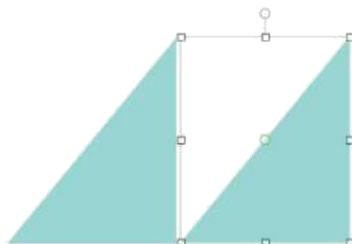
Hình 4.2.1. Vẽ hình chữ nhật

Sử dụng công cụ **Chỉnh hình** , nhấp vào một điểm ở góc trên bên trái để nó trở thành hình tam giác. Đây chính là một nửa của ngọn núi.



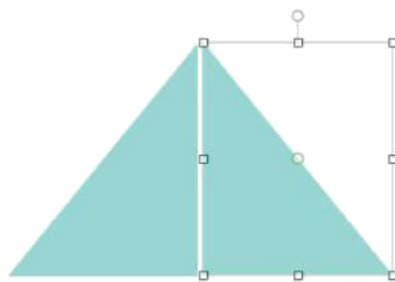
Hình 4.2.2. Vẽ nửa ngọn núi thứ nhất

Sử dụng công cụ , tạo thêm một bản sao để vẽ nửa còn lại.



Hình 4.2.3. Tạo bản sao nửa ngọn núi

Sau đó nhấp vào nút  để lật ngược hình thứ hai vừa sao.



Hình 4.2.4. Lật và sắp xếp lại nửa ngọn núi

Để ngọn núi có một bên chiếu sáng một bên bị che khuất, ta đổ màu tối hơn cho nửa bên phải.



Hình 4.2.5. Đổ màu cho nửa bên phải

Tiếp đó, để trên đỉnh núi có tuyết phủ trắng, ta vẽ một hình chữ nhật màu trắng rồi dùng công cụ **Chỉnh hình** để chỉnh thành đỉnh núi.



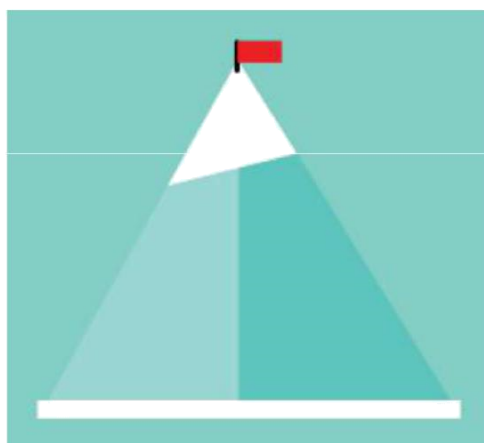
Hình 4.2.6. Vẽ tuyết trên đỉnh núi

Để vẽ lá cờ trên đỉnh núi, ta sử dụng công cụ **Vẽ đường thẳng** để vẽ cột cờ và công cụ **Vẽ hình chữ nhật** với màu nền màu đỏ để vẽ lá cờ.



Hình 4.2.7. Vẽ lá cờ

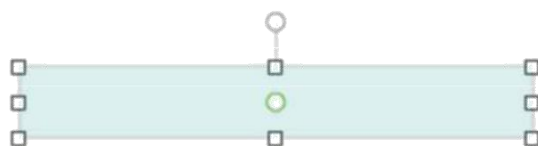
Cuối cùng, ta vẽ thêm một hình chữ nhật trắng dài dưới chân núi làm óp tuyết. Nhấn giữ phím Ctrl, dùng công cụ chọn để chọn các chi tiết của núi, sau đó gộp nhóm. Các bạn đã có một ngọn núi hoàn chỉnh.



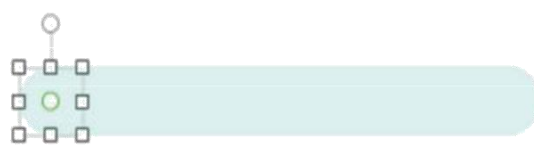
Hình 4.2.8. Ngọn núi hoàn chỉnh

2.2. Vẽ mây

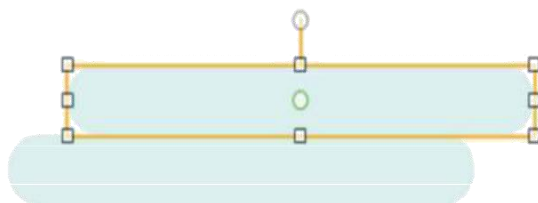
Mây của chúng ta khá đơn giản, gồm 2 hình chữ nhật và 4 hình tròn. Các bước thực hiện được mô tả ở Hình 4.2.9.



B1. Vẽ hình chữ nhật



B2. Vẽ thêm hai hình tròn ở hai đầu



B3. Tạo bản sao



B4. Sắp xếp lệch nhau

Hình 4.2.9. Các bước vẽ mây



Tóm tắt lí thuyết và bài tập thực hành

Vẽ núi và mây

Vẽ núi và mây

daylaptrinh.net

Núi

Mây

Bài tập

1. Thêm cây dưới chân núi.
2. Thêm một đám mây trên bầu trời.
3. Vẽ đối tượng là cờ ở vị trí đích đến của Nobita.

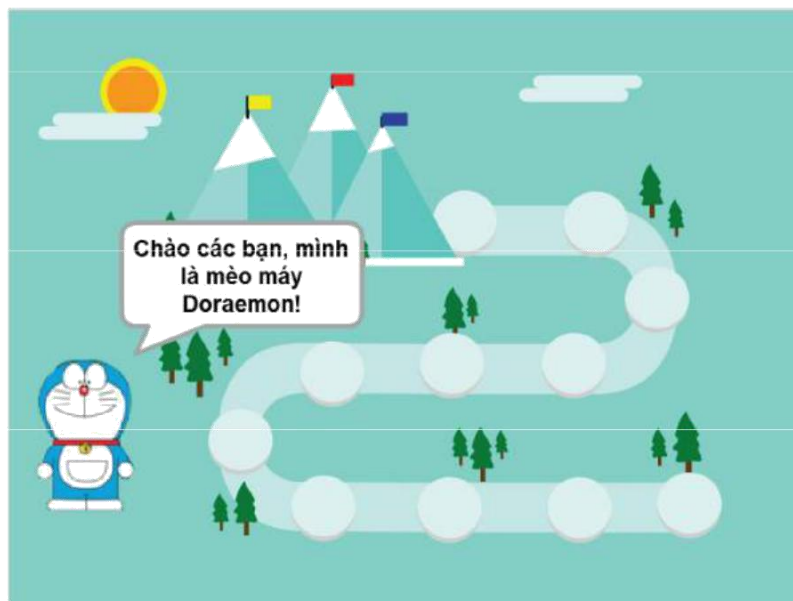
BÀI 3

LẬP TRÌNH CHO MÈO MÁY NÓI VÀ NGHỈ

3.1. Lập trình để đối tượng nói

Trước tiên, ta xoá nhân vật chú mèo có sẵn và tải tài liệu để thêm hình Doraemon vào Khung nhân vật.

Khi bắt đầu, chú mèo máy Doraemon sẽ giới thiệu về bản thân và chương trình.



Hình 4.3.1. Nhân vật nói

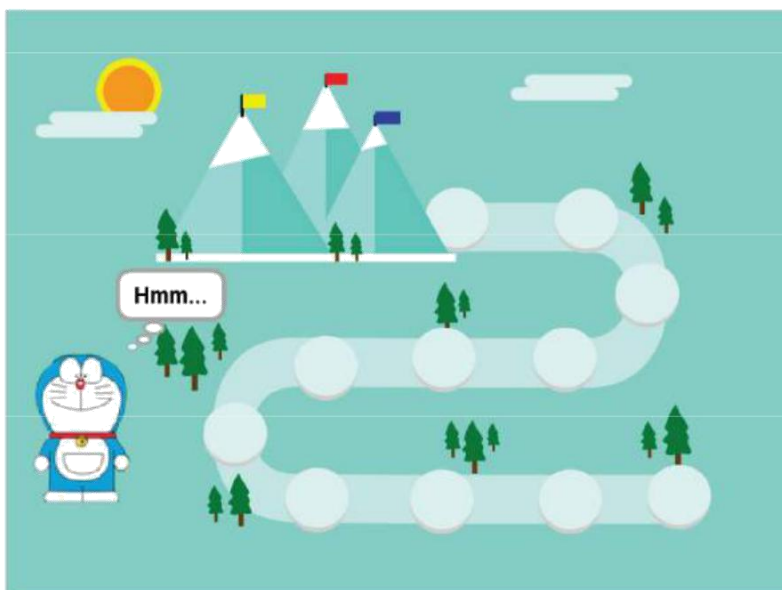
Để lập trình được cho Doraemon nói, ta sử dụng khối lệnh **nói** trong **2 giây** hoặc **nói**. Cả hai khối lệnh đều điều khiển đối tượng nói và có thể chỉnh sửa nội dung của lời nói tùy ý. Sự khác biệt nằm ở thời gian mà đối tượng nói.

Với khối lệnh **nói** trong **2 giây**, đối tượng sẽ nói trong một khoảng thời gian nhất định. Thời gian mặc định có sẵn trong khối lệnh là 2 giây, chúng ta có thể thay đổi lượng thời gian này tùy ý.

Với khối lệnh **nói**, nhân vật sẽ hiện lời nói mãi mãi cho đến khi có nói lời khác hoặc ta nhấp nút

3.2. Lập trình để đối tượng nghĩ

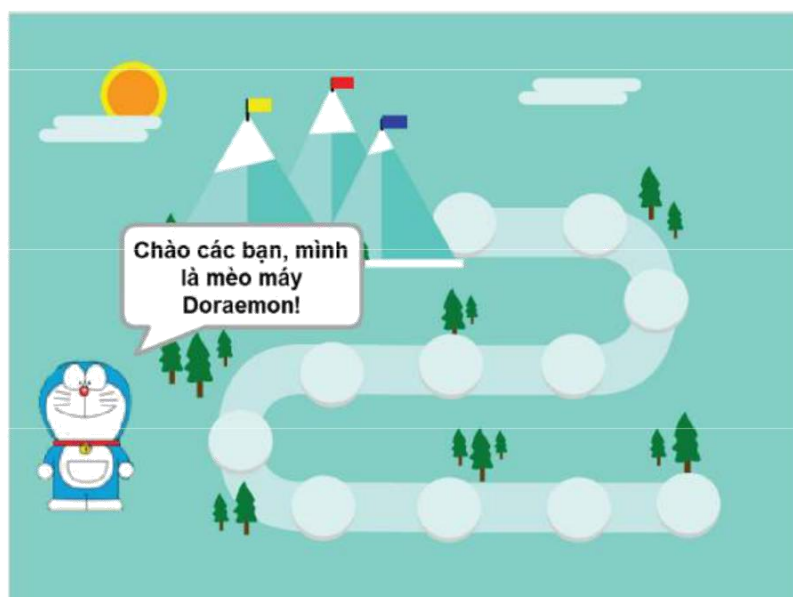
Để nhân vật nghĩ, chúng ta sử dụng hai khối lệnh **nghĩ** trong 2 giây và **nghĩ**. Sự khác nhau của hai khối lệnh này cũng giống khối lệnh nói.

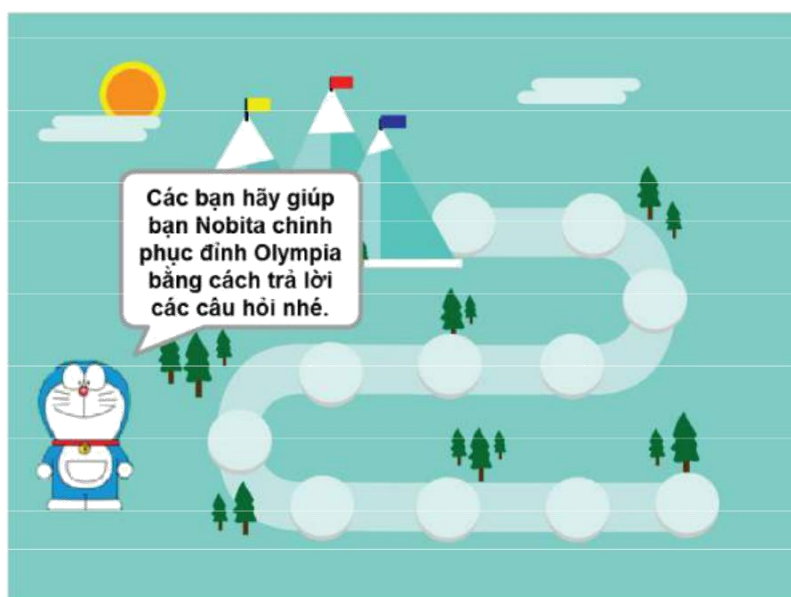


Hình 4.3.2. Nhân vật nghĩ

3.3. Lập trình cho Doraemon

Khi chạy chương trình, đầu tiên nhân vật Doraemon sẽ giới thiệu bản thân và chương trình trong một khoảng thời gian ngắn. Chúng ta sẽ sử dụng khối lệnh **nói** trong 2 giây. Hình 4.3.5 là một ví dụ mẫu cho lời giới thiệu.





khí nhấn vào

nói Chào các bạn, mình là mèo máy Doraemon! trong 5 giây

nói Các bạn hãy giúp bạn Nobita chinh phục đỉnh Olympia bằng cách trả lời các câu hỏi nhé. trong 5 giây

Hình 4.3.3. Ví dụ Doraemon giới thiệu



Tóm tắt lí thuyết và bài tập thực hành

Nói và nghĩ

Lập trình cho Doraemon có thể nói và nghĩ

daylaptrinh.net

Nói

Xin chào các bạn !

nói Xin chào các bạn ! trong 5 giây

Nghĩ

Hmm...

nghĩ Hmm...

Bài tập

1. Chỉnh sửa thời gian nói lâu hơn.
2. Tự thu âm và thêm âm thanh chạy cùng đoạn thoại của Doraemon.
3. Thêm trạng thái thứ hai cho Doraemon.



BÀI 4

GIỚI THIỆU VỀ BIẾN

Biến là một khái niệm rất quan trọng. Trong bài này, các bạn sẽ được tìm hiểu về cách tạo biến và một số khối lệnh thao tác với biến trong Scratch.

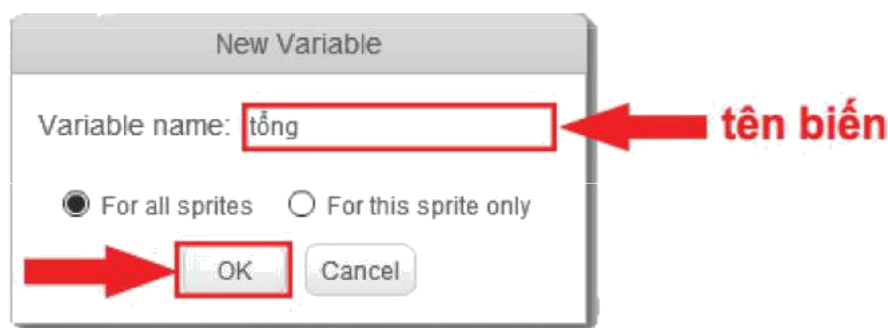
4.1. Biến trong Scratch

Biến là một đại lượng được đặt tên dùng để lưu trữ số hoặc kí tự và có thể thay đổi được. Trong Scratch, biến là một khối lệnh có hình dạng , chứa một giá trị nhất định có thể là một số, một kí tự hoặc một chuỗi các kí tự (chuỗi kí tự).



Hình 4.4.1. Các biến khác nhau với các giá trị của chúng

Tất cả các thao tác với biến ở nhóm **Dữ liệu**. Tạo một biến mới bằng cách nhấp vào nút **Make a Variable** trong nhóm **Dữ liệu** và một cửa sổ sẽ hiện ra.



Hình 4.4.2: Cửa sổ tạo biến mới

Ta nhập tên biến vào ô *Variable name*. Tên biến nên có ý nghĩa khơi gợi chức năng của biến. Ví dụ như nếu biến dùng để chứa giá trị tổng của hai số thì nên đặt tên là **Tổng**, biến để lưu điểm thì nên đặt tên là **Điểm**.

Trong cửa sổ tạo biến có hai lựa chọn. Lựa chọn *For this sprite only* để biến chỉ được thao tác, sử dụng bởi đối tượng hiện tại đang lập trình. Lựa chọn *For all sprites* để biến được tạo ra có thể được thao tác bởi tất cả các nhân vật khác trong chương trình.

Nhấp nút **OK** để tạo biến. Trong ví dụ này, một biến tên là **Tổng** được tạo ra để tất cả các nhân vật đều có thể sử dụng được.





Hình 4.4.3. Ví dụ biến **Tổng**

Sau khi tạo biến, những khối lệnh giúp thao tác với biến sẽ xuất hiện trên bảng danh mục các khối lệnh:

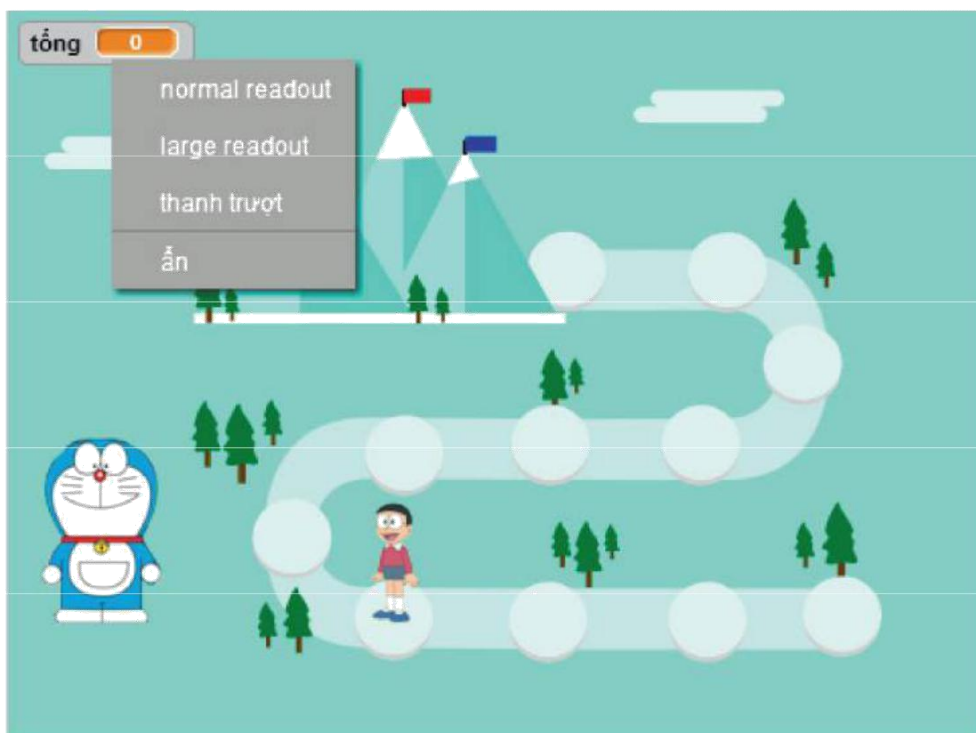
đặt tổng là 0: Đặt cho biến **Tổng** giá trị bất kì. Trong trường hợp này giá trị được đặt cho biến **Tổng** là 0.

thay đổi tổng một lượng 1: Thay đổi biến **Tổng** một lượng bất kì. Trong trường hợp này lượng giá trị thay đổi của biến **Tổng** là 1.

hiện biến tổng: Hiển thị biến **Tổng** lên trên sân khấu.

ẩn biến tổng đi: Ẩn biến **Tổng** đi

Khi hiển thị trên sân khấu, biến có ba kiểu để hiển thị. Để chuyển đổi kiểu hiển thị, ta nhấp đúp chuột lên khung hiển thị của biến trên sân khấu, hoặc nhấp nút phải chuột vào khung hiển thị và chọn cách hiển thị cho biến.



Hình 4.4.4. Chuyển đổi kiểu hiển thị biến

Các kiểu hiển thị:

0 : Chỉ hiển thị giá trị của biến.

tổng 0 : Hiển thị tên của biến cùng với giá trị.

tổng 0 : Hiển thị tên biến, giá trị của biến và thanh trượt. Các bạn có thể dùng thanh trượt để thay đổi giá trị của biến.



Tóm tắt lý thuyết và bài tập thực hành

Giới thiệu về biến

Làm quen với khái niệm Biến trong Scratch

Chuyển động

Ngoại hình

Âm thanh

Đồ vẽ

Dữ liệu

Sự kiện

Điều khiển

Cảm biến

Phép toán

Thêm nhiều khối nữa

Make a Variable

☒ **tổng**

đặt: tổng là 0

thay đổi: tổng một lượng 1

hiện biến: tổng

ẩn biến: tổng

Make a List

daylaptrinh.net

Kiểu giá trị

2
a

Xin chào
lời nói

@
ký tự

Kiểu hiển thị

0

tổng 0

tổng

0

Bài tập

Lập trình để biến ẩn/hiện trên sân khấu

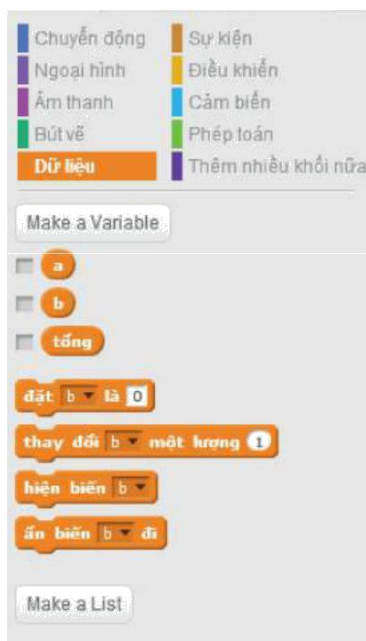


BÀI 5

LẬP TRÌNH CHO DORAEMON TÍNH TOÁN

5.1. Tính tổng hai số ngẫu nhiên

Một phép tính tổng của hai số gồm hai số hạng và tổng. Chúng ta đã có biến để chứa giá trị của tổng từ bài trước, vì thế nên ta chỉ cần tạo thêm hai biến a và b để lưu trữ giá trị của hai số hạng.



Hình 4.5.1. Thêm hai biến lưu hai số hạng

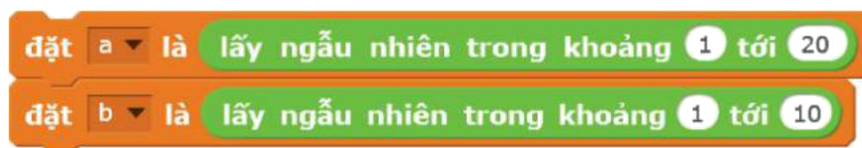
Doraemon sẽ đặt ra câu hỏi là tính tổng hai số ngẫu nhiên. Hai số hạng đều được lấy ngẫu nhiên, tức là chúng ta phải đặt cho biến a và biến b giá trị ngẫu nhiên. Để lấy giá trị ngẫu nhiên, chúng ta sẽ cần đến khối lệnh **lấy ngẫu nhiên trong khoảng 1 tới 20** đã được hướng dẫn ở Bài 6 trong Chương 3.

Ta kéo khối lệnh đặt biến sang, nhấp vào mũi tên để chọn biến a và sau đó kéo khối lệnh **lấy ngẫu nhiên trong khoảng 1 tới 20** vào **đặt a là** để đặt cho biến a giá trị ngẫu nhiên.



Hình 4.5.2. Đặt cho biến a giá trị ngẫu nhiên

Tạo bản sao và thay đổi biến a thành biến b , sau đó ghép hai khối lệnh lại với nhau như Hình 4.5.3.



Hình 4.5.3. Đặt hai biến giá trị ngẫu nhiên

Sau khi đã có hai số hạng, chúng ta sẽ lập trình để tính tổng hai số hạng này và lưu vào biến **Tổng**. Chúng ta sử dụng khối lệnh $\text{a} + \text{b}$ để tính tổng hai số. Hình 4.5.4 mô tả cách sử dụng khối lệnh này.



Hình 4.5.4. Khối lệnh tính tổng

Việc cuối cùng là đặt giá trị biến **Tổng** là giá trị của khối lệnh $\text{a} + \text{b}$ và ghép vào đoạn khối lệnh bên trên.



Hình 4.5.5. Đoạn khối lệnh tính toán



Tóm tắt lí thuyết và bài tập thực hành

Doraemon tính toán

Lập trình để Doraemon tính tổng hai số ngẫu nhiên

daylaptrinh.net

Lấy ngẫu nhiên

lấy_ngẫu_nhiên_trong_khoảng 1 tới 20

Khởi lệnh

đặt a là lấy ngẫu nhiên trong khoảng 1 tới 20

đặt b là lấy ngẫu nhiên trong khoảng 1 tới 10

đặt tổng là a + b

Bài tập

1. Lập trình để biến a lấy ngẫu nhiên từ 1 đến 50
2. Lập trình để biến b lấy ngẫu nhiên từ 1 đến 50
3. Tạo thêm biến có tên vị trí.

BÀI 6 LẬP TRÌNH ĐỂ DORAEMON ĐẶT CÂU HỎI

Sau khi Doraemon đã tính toán xong, chú mèo máy sẽ đặt ra câu hỏi về phép tính tổng vừa được tính.

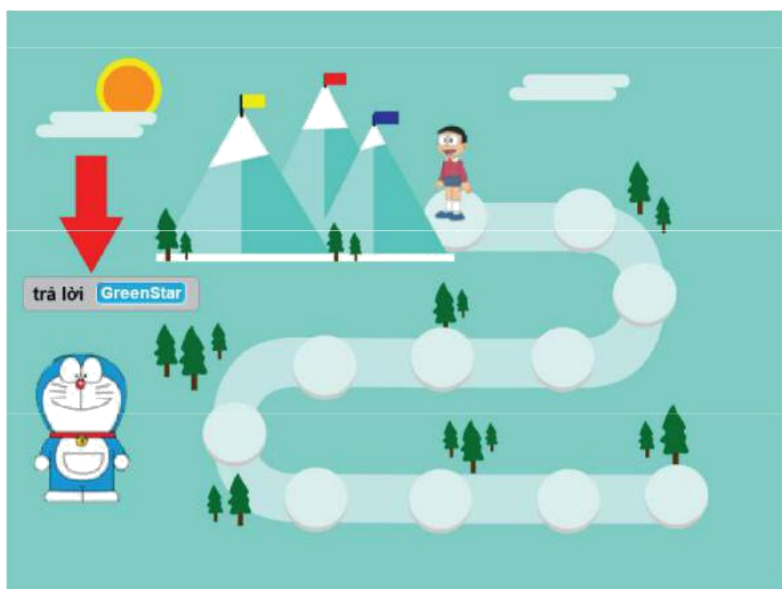
6.1. Khối lệnh hỏi

Để Doraemon đặt câu hỏi, ta sử dụng khối lệnh **hỏi** `What's your name?` và **đợi** trong nhóm **Cảm biến**. Mặc định nội dung câu hỏi là “What's your name?”. Nếu chạy thử thì ta sẽ thấy Doraemon hỏi và trên sân khấu xuất hiện thêm một khung để nhập câu trả lời.



Hình 4.6.1. Khối lệnh hỏi

Chỉ khi nào chúng ta nhập câu trả lời và nhấn phím Enter hoặc nhấp nút thì Doraemon mới ngừng hỏi và câu trả lời vừa nhập vào sẽ được lưu lại ở khối lệnh **trả lời** trong nhóm **Cảm biến**. Ta có thể chạy khối lệnh để xem lại câu trả lời được lưu, hoặc nhấp vào ô tích bên cạnh ☐ **trả lời** để chuyển thành ☒ **trả lời** và giá trị khối lệnh lưu sẽ hiển thị trên sân khấu.



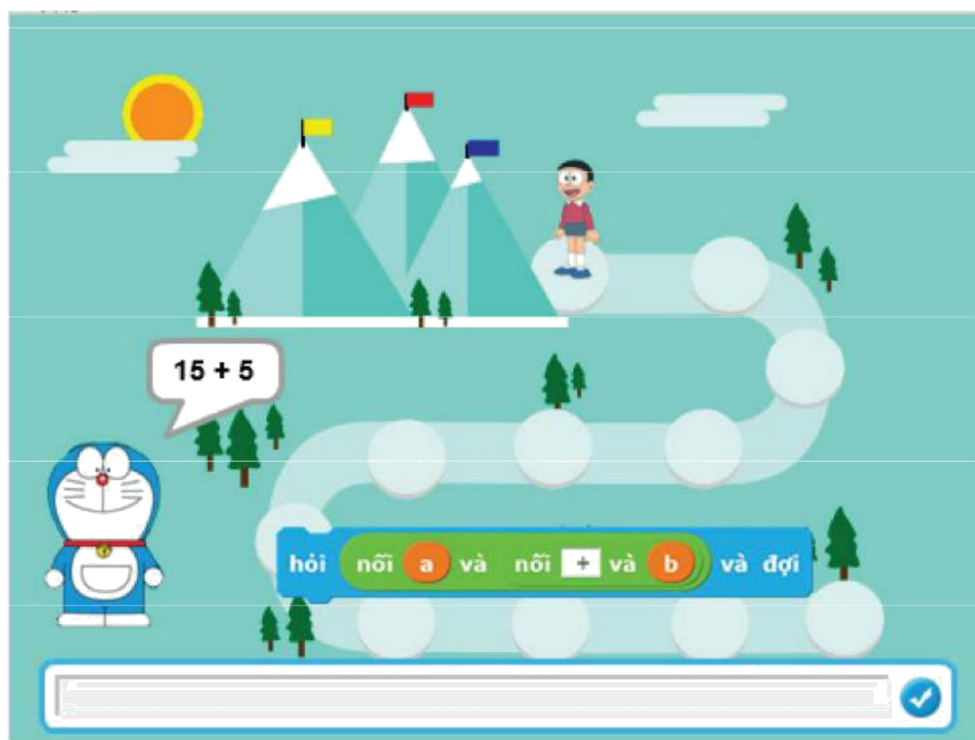
Hình 4.6.2. Giá trị câu trả lời được lưu

Nội dung câu hỏi mà Doraemon hỏi là phép toán tính tổng hai biến a và b . Nếu ghép khối lệnh **a + b** vào thành nội dung câu hỏi thì Doraemon nói ra kết quả của tổng chứ không phải đưa ra phép toán, vậy nên chúng ta sẽ phải sử dụng khối lệnh **nói [] và []**. khối lệnh này sẽ có giá trị đặc định là nói "hello" và "world", các bạn chạy thử sẽ thấy cách khối lệnh hoạt động.



Hình 4.6.3. Khối lệnh nói

Trong trường hợp này, các bạn cần nối giá trị biến a , dấu ‘+’ và giá trị biến b lại, tổng cộng là ba ô nối. Chúng ta ghép nối các khối lệnh thành **nối và nối và** , sau đó ghép biến a và biến b vào hai bên và điền dấu ‘+’ vào giữa, ta được **nối và nối và** .



Hình 4.6.4. Kết quả



Tóm tắt lí thuyết và bài tập thực hành

Lập trình để Doraemon hỏi

Khởi lệnh hỏi, trả lời và khởi lệnh nói

daylaptrinh.net

Nhân vật

Khởi lệnh

hỏi nói a và nói + và b và đợi

Bài tập

1. Lập trình để khi Doraemon hỏi xong, chúng ta nhập câu trả lời thì Doraemon sẽ hỏi tiếp.
2. Lập trình để đặt giá trị biến vị trí bằng 0.
3. Thêm nhân vật Nobita vào chương trình, lập trình cho Nobita nhảy tới vị trí bắt đầu.

BÀI 7 LẬP TRÌNH ĐỂ DORAEMON KIỂM TRA CÂU TRẢ LỜI

Khi Doraemon đặt câu hỏi, người chơi sẽ phải nhập câu trả lời vào và Doraemon sẽ kiểm tra câu trả lời, nếu đúng thì sẽ cho Nobita đến gần đích hơn, nếu sai thì sẽ nói “Sai rồi !”. Bài học này sẽ hướng dẫn các bạn lập trình cho Doraemon thực hiện điều đó.

7.1. Hai khối lệnh kiểm tra điều kiện

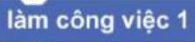
Nhóm **Điều khiển** có hai khối lệnh để kiểm tra điều kiện:



Cả hai khối lệnh đều có ô chứa khối lệnh giá trị hình lục giác  (còn gọi là khối lệnh điều kiện, chứa khối lệnh dạng ).



Hình 4.7.1. Khối lệnh Nếu – thì

Với khối lệnh **Nếu – thì**, nếu khối lệnh  chứa giá trị **true** (đúng) thì đoạn khối lệnh  sẽ được thực hiện, nếu không thì bỏ qua và tiếp tục thực hiện khối lệnh tiếp theo .



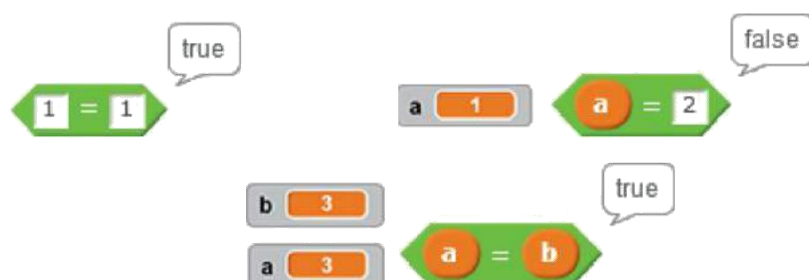
Hình 4.7.2. Khối lệnh Nếu – thì – còn không thì



Với khối lệnh **Nếu – thì – còn không thì**, nếu khối lệnh điều kiện **điều kiện** chứa giá trị đúng thì khối lệnh **làm công việc 1** sẽ được thực hiện và bỏ qua khối lệnh **làm công việc 2**. Nếu khối lệnh **điều kiện** chứa giá trị sai thì sẽ ngược lại, bỏ qua khối lệnh **làm công việc 1** và thực hiện khối lệnh **làm công việc 2**. Sau đó sẽ đều tiếp tục thực hiện các khối lệnh tiếp theo.

7.2. Kiểm tra câu trả lời

Doraemon sẽ kiểm tra xem câu trả lời có đúng với đáp án hay không, tức là sẽ phải so sánh đáp án với câu trả lời nhập vào có bằng nhau không. Để so sánh hai giá trị có bằng nhau hay không, chúng ta sử dụng khối lệnh **□ = □**.



ình 4.7.3. hối lệnh so sánh

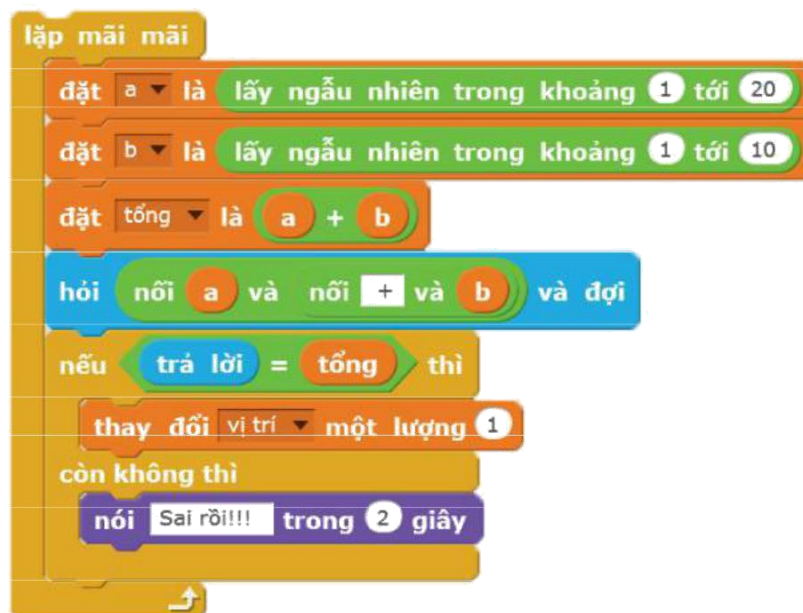
Biến **Tổng** lưu đáp án còn khối lệnh **trả lời** lưu câu trả lời. Để so sánh hai giá trị này với nhau, các bạn ghép thành **trả lời = tổng**.

Nếu trả lời đúng, **trả lời** bằng với **tổng**, thì sẽ đưa Nobita đến gần đích hơn bằng cách thay đổi giá trị biến **Vị trí**. Chúng ta sử dụng biến **vị trí** đã được thêm vào ở phần bài tập của bài trước để lưu vị trí của Nobita. Khi trả lời đúng thì vị trí của Nobita sẽ được tăng lên một bậc, các bạn lập trình cho biến **Vị trí** tăng thêm 1 bằng khối lệnh **thay đổi vị trí ▾ một lượng 1**. Còn nếu sai thì Doraemon sẽ nói “Sai rồi!”.

Hình 4.7.4. Kiểm tra câu trả lời



Sau khi trả lời xong, Doraemon sẽ hỏi lại câu hỏi khác, việc Doraemon đặt câu hỏi và kiểm tra câu trả lời sẽ được lặp lại mãi mãi.



Hình 4.7.5. Đoạn khối lệnh hoàn chỉnh



Tóm tắt lí thuyết và bài tập thực hành

Kiểm tra câu trả lời

Lập trình để Doraemon kiểm tra câu trả lời của người chơi.

daylaptrinh.net

Kiểm tra điều kiện

nếu thì
 nếu thì
 còn không thì

Khối lệnh

```

nếu trả lời = tổng thì
  thay đổi vị trí một lượng 1
còn không thì
  nói Sai rồi!!! trong 2 giây
  
```

Bài tập

1. Thêm âm thanh trả lời đúng và sai vào nhân vật mèo máy (âm thanh có trong tài liệu của chương trình).
2. Lập trình để khi trả lời đúng thì âm thanh đúng sẽ được chạy, trả lời sai thì âm thanh sai sẽ được chạy.



BÀI 8

LẬP TRÌNH CHO NOBITA

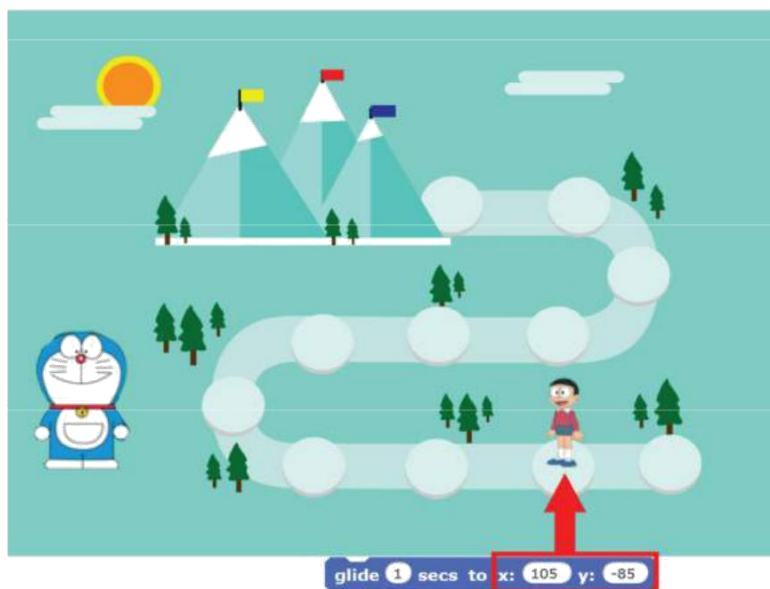
8.1. Khối lệnh Glide

Khối lệnh `glide 1 secs to x: -29 y: 19` là khối lệnh thuộc nhóm **Chuyển động**. Khối lệnh này giúp điều khiển nhân vật trượt tới vị trí nhất định trong một khoảng thời gian nhất định.



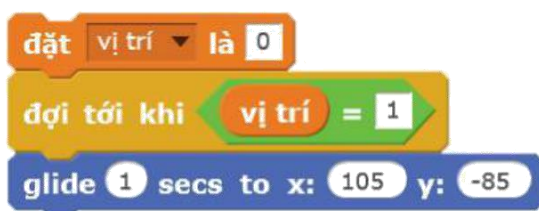
Hình 4.8.1. Khối lệnh Glide

Khi bắt đầu chương trình, Nobita sẽ đứng ở vị trí đầu tiên và việc di chuyển đến vị trí tiếp theo chúng ta sẽ sử dụng khối lệnh **Glide**.



Hình 4.8.2. Nobita di chuyển

Ta chọn nhân vật Nobita, vào Khu lập trình, khởi tạo biến **vị trí** được đặt bằng 0, Nobita sẽ phải đợi cho đến khi người chơi nhập vào câu trả lời đúng, biến **Vị trí** tăng thêm 1 thì mới di chuyển đến vị trí thứ hai. Các bạn cần sử dụng khối lệnh **đợi tới khi** trong nhóm **Điều khiển** để làm việc này.



Hình 4.8.3. Đoạn khối lệnh ứng với vị trí đầu tiên

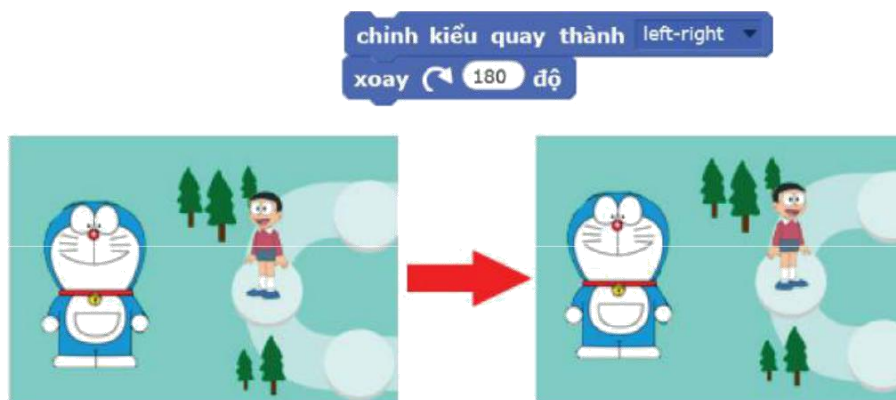
Để từ vị trí thứ hai Nobita di chuyển lên vị trí thứ ba, ta cần xác định được tọa độ vị trí thứ ba, đồng thời phải đợi cho đến khi người chơi trả lời đúng câu hỏi tiếp theo (biến **Vị trí** tăng thêm 1 thành 2) mới tiến hành di chuyển. Tương tự với các vị trí tiếp theo.



Hình 4.8.4. Đoạn khối lệnh ứng với vị trí thứ hai

Đến vị trí thứ tư, Nobita phải xoay người hướng ngược lại (hướng sang phải), do đó các bạn điều khiển Nobita xoay một góc 180° bằng khối lệnh **xoay 180 độ**.

Khi chạy thử, Nobita bị quay ngược hướng dư i. Để nhân vật không bị lật, ta sử dụng khối lệnh **chỉnh kiểu quay thành left-right**.



Hình 4.8.5. Thay đổi kiểu xoay của nhân vật



Tóm tắt lí thuyết và bài tập thực hành

Lập trình cho Nobita

Di chuyển và xoay nhân vật

daylaptrinh.net

Hành động

Di chuyển

Xoay

Khởi lệnh

Bài tập

1. Lập trình để mỗi khi vị trí được tăng lên thì nobita sẽ trượt tới vị trí đó.
2. Thêm hình ảnh chiến thắng cho Nobita, khi đến được điểm cuối sẽ đổi hình ảnh.
3. Lập trình để khi bắt đầu chương trình, Nobita sẽ đổi về hình ảnh bình thường.

BÀI 9

KẾT THÚC CHƯƠNG TRÌNH

Bài học cuối cùng trong chương sẽ hướng dẫn các bạn hoàn thiện chương trình và cách dừng chương trình để kết thúc khi Nobita đã đến đích.

9.1. Hoàn thiện chương trình

Mỗi khi trả lời đúng, Nobita chưa kịp di chuyển lên vị trí mới Doraemon đã hỏi câu hỏi tiếp theo. Chúng ta lập trình để Doraemon đợi thêm một khoảng thời gian cho Nobita di chuyển rồi mới hỏi bằng khối lệnh **đợi 2 giây**. Ta thêm khối lệnh với nhân vật Doraemon như sau:



Hình 4.9.1. Hiệu chỉnh Doraemon

Khi Nobita đến đích, chương trình chưa được dừng lại và Doraemon vẫn còn hỏi tiếp. Để kết thúc chương trình, ta sử dụng khối lệnh **ngừng tất cả** trong nhóm **Điều khiển**. Ta thay đổi lệnh với nhân vật Nobita như sau:



Hình 4.9.2. Ngừng toàn bộ chương trình

Chương trình đã được hoàn thiện, bây giờ chúng ta có thể lập trình một trò chơi cho riêng mình.





Tóm tắt lí thuyết và bài tập thực hành

Kết thúc chương trình

Hoàn thiện chương trình

daylaptrinh.net

Nhân vật

Khối lệnh

```

nếu trả lại = tổng thì
  thay đổi vị trí một hướng 1
  đợi 2 giây
  còn không thì
    nói "Sạc rồi!!" trong 2 giây
    đợi tới khi vị trí = 10
    glide 1 secs to x: 38 y: 85
    ngừng tất cả
        
```

Bài tập

- Lập trình thêm tuyết bay từ trên xuống dưới.
- Đổi màu cho bông tuyết thành màu trắng.

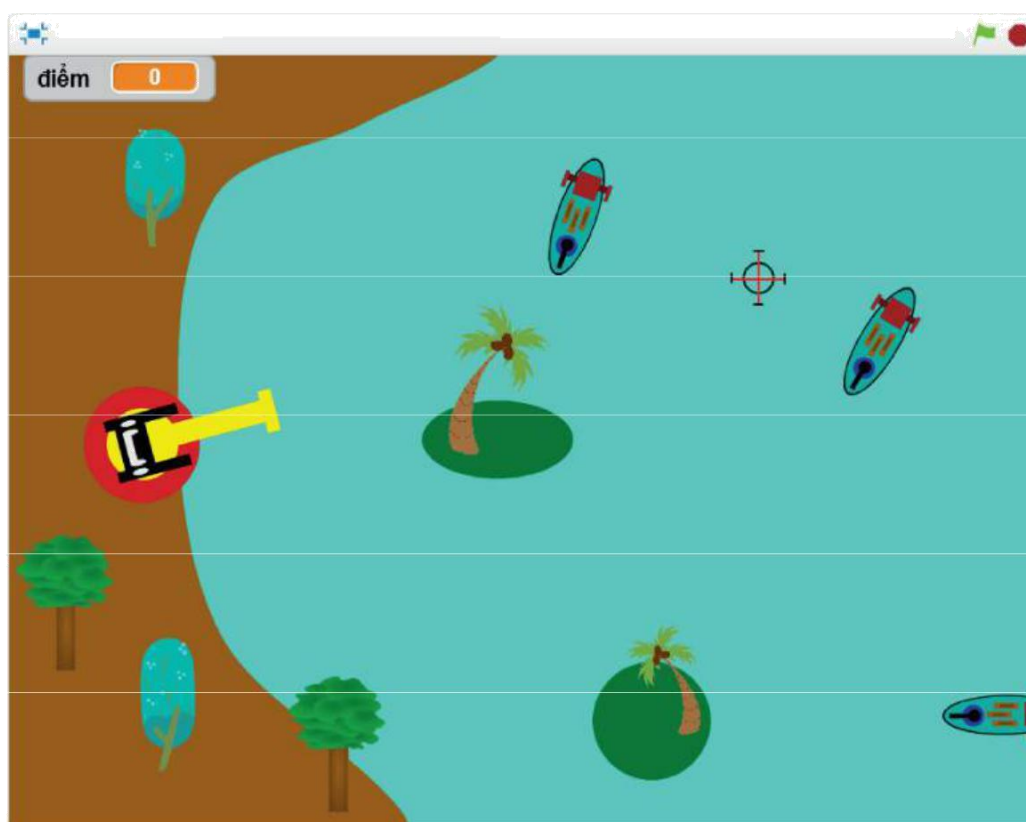
CHƯƠNG



Bảo vệ đảo

Sản phẩm mà các bạn được hướng dẫn trong chương này là một trò chơi. Các bạn sẽ đóng vai các chiến sĩ, điều khiển khẩu pháo để bắn hạ các tàu địch đang muốn tiến tới xâm chiếm hai hòn đảo của chúng ta.

Tổng quan chương trình



Trong trò chơi, các tàu địch sẽ xuất hiện ngẫu nhiên trên phần vùng biển của bản đồ và di chuyển đến hai đảo. Người chơi sẽ phải sử dụng chuột để bắn các tàu đó bằng cách nhấp nút trái chuột vào tàu muốn bắn. Mỗi khi bắn trúng, tàu địch sẽ bị hạ và người chơi sẽ được cộng điểm. Nếu tàu địch chạm tới một trong hai đảo, trò chơi sẽ kết thúc.



Chương trình gồm có các đối tượng:

1. Đảo 1

Đảo thứ nhất cần bảo vệ.



2. Đảo 2

Đảo thứ hai cần bảo vệ.



3. Pháo

Pháo nằm cố định tại một vị trí trên đất liền và luôn quay nòng về phía con trỏ chuột.



4. Lửa

Lửa luôn được ẩn và chỉ xuất hiện trong một thời gian ngắn khi người chơi nháy nút trái chuột để bắn.



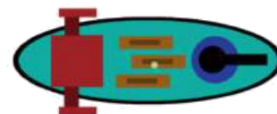
5. Tâm ngắm

Tâm ngắm luôn di chuyển theo con trỏ chuột và rùng với con trỏ chuột.



6. Tàu địch

Tàu địch có vị xuất phát ngẫu nhiên từ



biên bên trên hoặc biên bên phải của bản đồ và tiến về một trong hai đảo.

7. Chữ Start

Hiện lên khi nháy vào nút lệnh lá cờ xanh rồi

lại ẩn đi, báo hiệu trò chơi được bắt đầu.

Start

8. Chữ Game Over

Game Over

Được ẩn đi, đến khi Tàu địch chạm vào một trong hai đảo thì mới hiện lên, báo hiệu trò chơi kết thúc.

Không phải tất cả các chức năng của chương trình sẽ được hướng dẫn mà sẽ có một số phần được đưa vào phần Bài tập thực hành ở cuối mỗi bài. Các bạn cần làm bài tập đầy đủ để hoàn thiện chương trình.

BÀI 1

VẼ BẢN ĐỒ

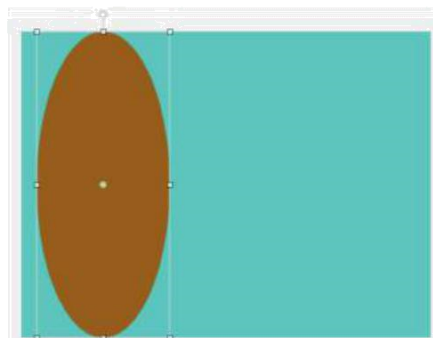
Trước tiên, chúng ta xoá **Chú mèo** đi và vẽ bản đồ làm hình nền cho chương trình.

1.1 Vẽ đất liền

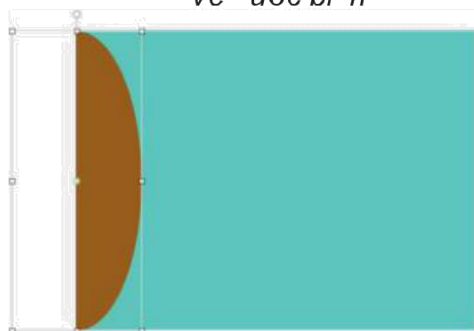
Chọn khung nền sân khấu vào Khu thiết kế phong nền, chuyển về chế độ Vector, vẽ một hình chữ nhật màu xanh nước biển kín trang vẽ. Phần đất liền được vẽ bằng hình elip màu nâu, chỉnh vị trí của hình ra bên trái và sử dụng công cụ Chỉnh hình như các bước ở Hình 5.1.1.



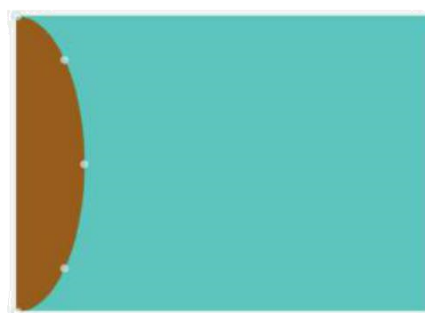
Vẽ ước bi n



Vẽ hình elip



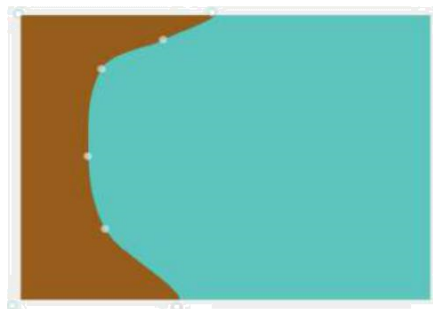
Đưa hình Elip sang lề trái



Dùng công cụ Chỉnh hình



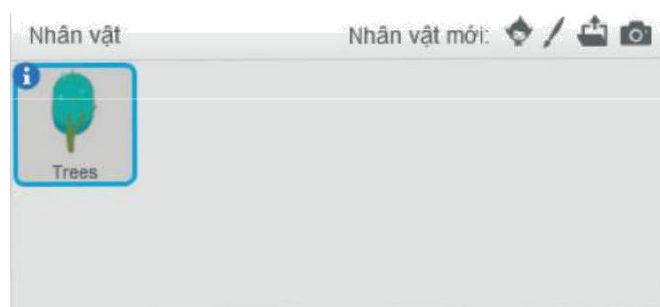
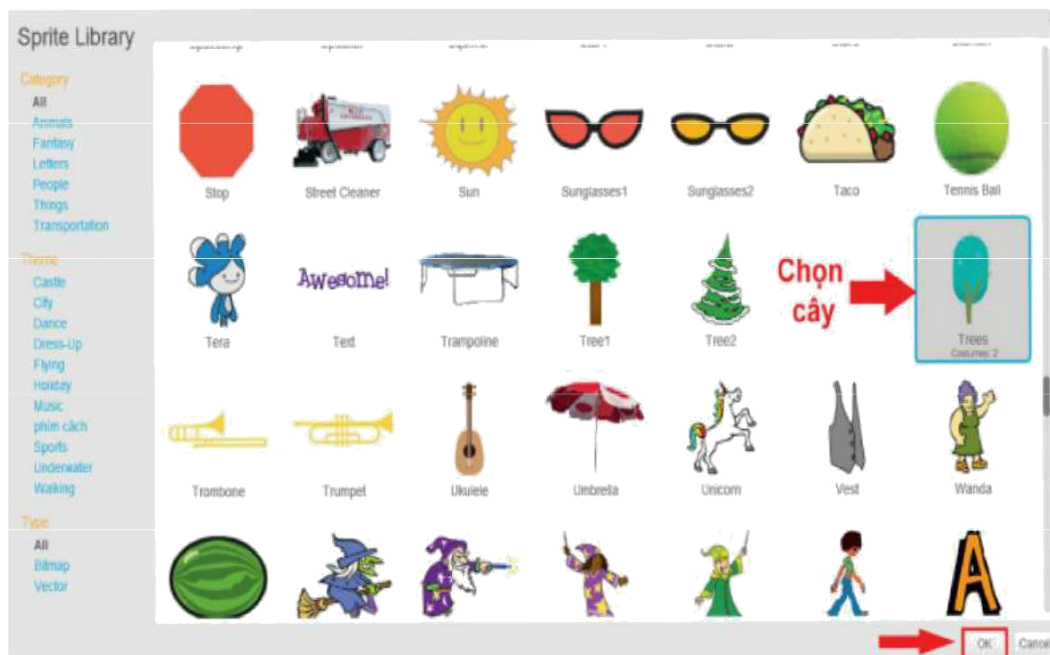
Chỉnh lại đường biên



Hình 5.1.1. Vẽ biển và đất liền

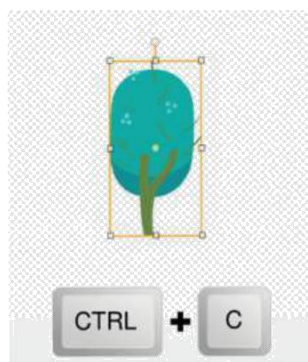


Thêm đối tượng cây từ thư viện ở Khung nhân vật.



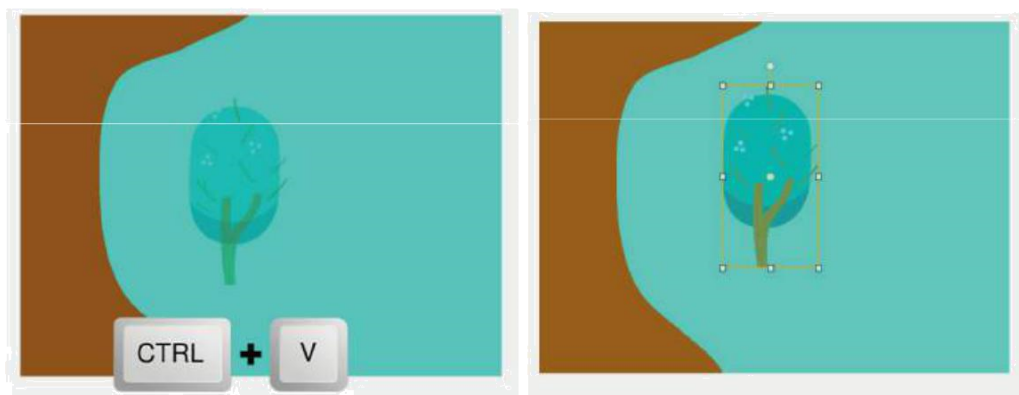
Hình 5.1.2. Thêm nhân vật cây

Để hình nền có hình ảnh cây, nhân vật cây chuyển sang khu ngoại hình (Costume), chọn cây và nhấn tổ hợp phím Ctrl + C để sao chép.



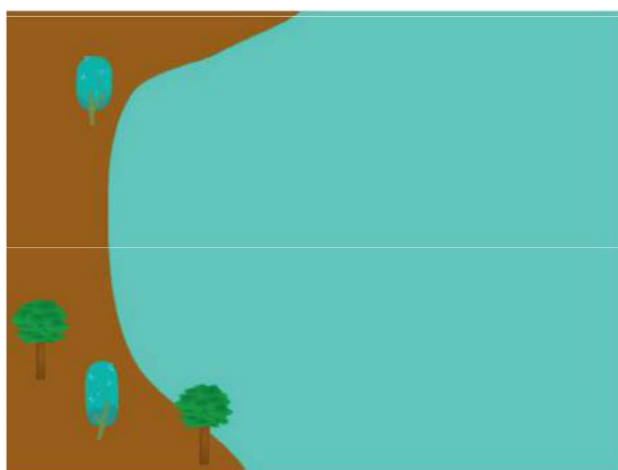
Hình 5.1.3. Sao chép hình ảnh cây

Chọn hình nền ở khu thiết kế phong nền, nhấn tổ hợp phím Ctrl + V, nháy chuột để dán hình cây và thay đổi kích thước cho hợp lí.



Hình 5.1.4. Dán hình và thay đổi kích thước

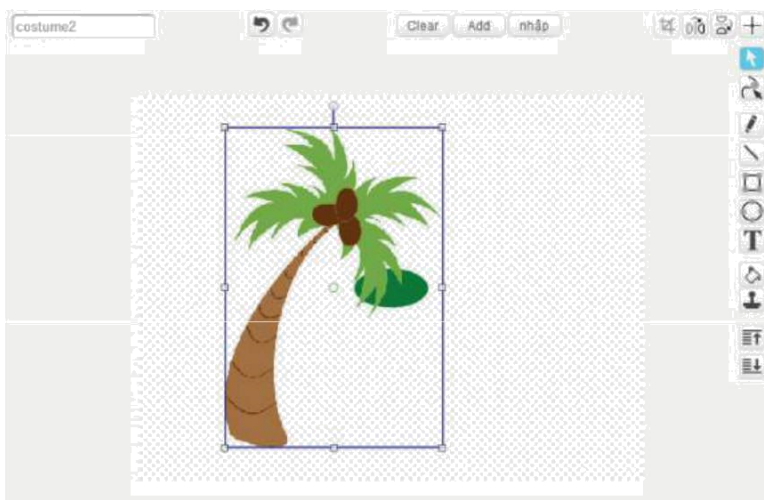
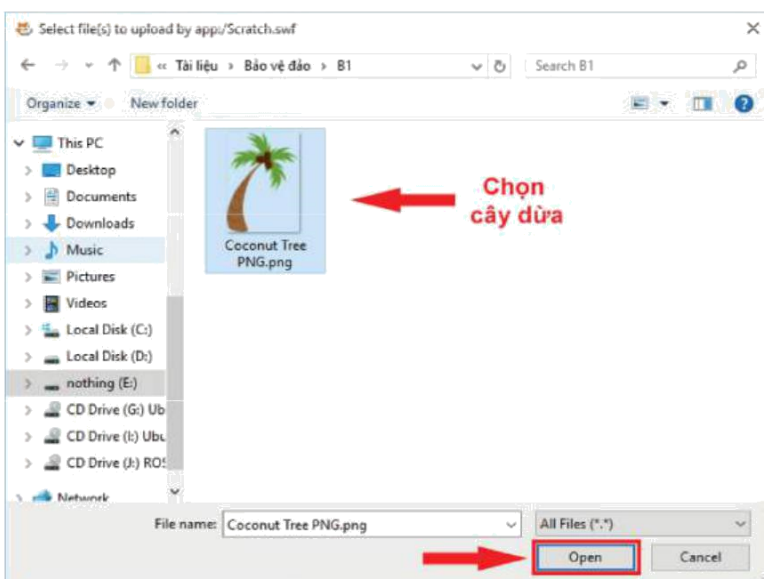
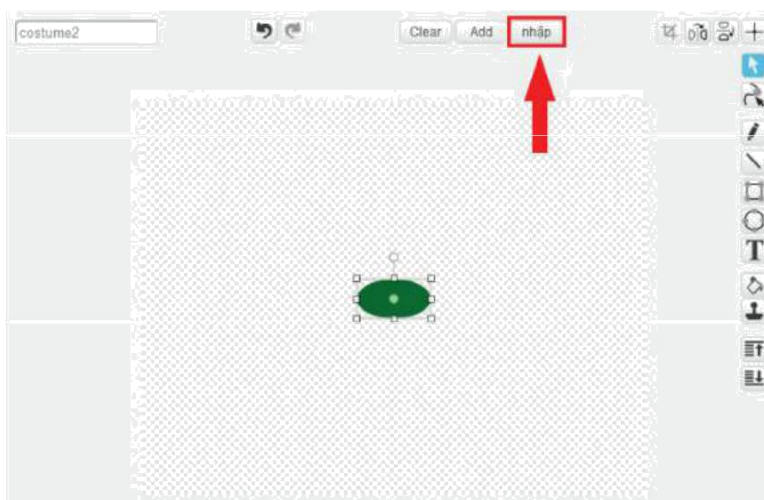
Ta có thể thêm nhiều cây khác bằng cách tương tự. Sau khi thêm xong ta chú ý xóa nhân vật Cây ở Khung nhân vật.



Hình 5.1.5. Đất liền hoàn chỉnh

1.2. Vẽ hai đảo

Hai đảo là hai đối tượng riêng biệt, nên ta cần nháy vào biểu tượng  ở khung nhân vật để vẽ thêm đối tượng. Đầu tiên vẽ **Đảo 1** bằng hình elip. Sau đó ta thêm vào cây dừa có sẵn trong tài liệu tải về bằng nút . Chỉnh kích thước cây dừa cho phù hợp.





Hình 5.1.6. Vẽ Đảo 1

Đảo 2 được vẽ tương tự, nhưng vẽ hình tròn thay cho hình elip và sử dụng công cụ



để lật cây dừa theo hướng ngược lại.



Hình 5.1.7. Vẽ Đảo 2



Tóm tắt lí thuyết và bài tập thực hành

Vẽ Bản Đồ

Vẽ đất liền và hai đảo

daylaptrinh.net

Đất liền

Đảo

Bài Tập

1. Lập trình cho hai đảo nhảy tới đúng vị trí trên bản đồ.
2. Đổi lại tên cho 2 đối tượng Đảo 1 và Đảo 2.
3. Vẽ tâm ngắm và chỉnh tâm cho tâm ngắm.

BÀI 2

VẼ PHÁO VÀ TÀU ĐỊCH

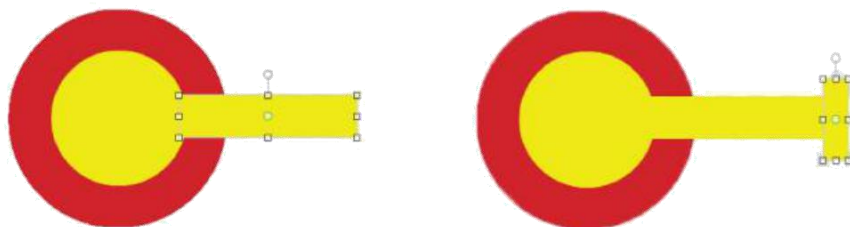
2.1. Vẽ Pháo

Ta thêm đối tượng mới bằng cách nháy vào biểu tượng , chuyển hình ảnh ở chế độ Vector và tiến hành vẽ pháo. Đầu tiên là vẽ thân pháo và tháp pháo bằng 2 hình tròn. Ta nhấn giữ phím Shift khi vẽ để vẽ được hình tròn.



Hình 5.2.1. Vẽ thân pháo

Tiếp theo là vẽ nòng pháo bằng hai hình chữ nhật.



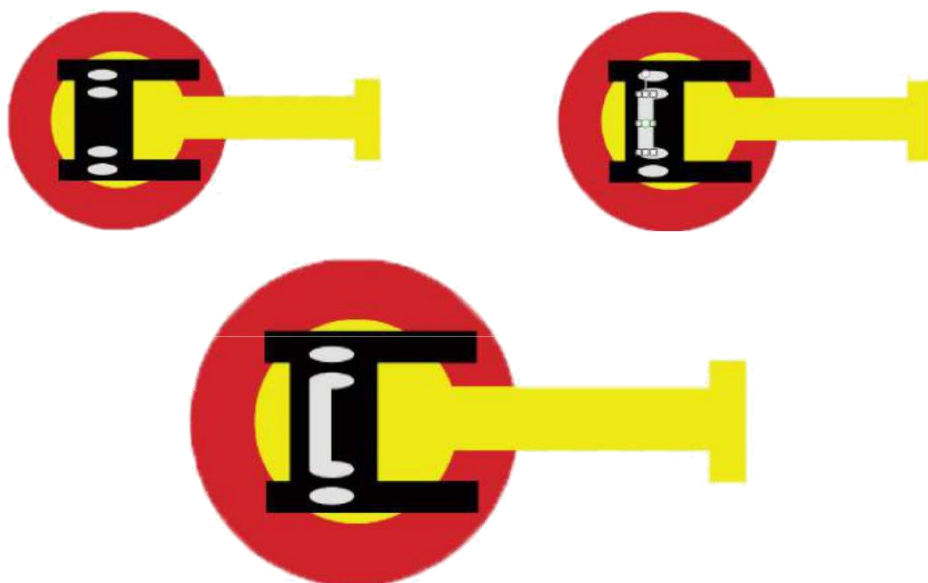
Hình 5.2.2. Vẽ nòng pháo

Hai súng bên trên tháp pháo sẽ được vẽ bằng ba hình chữ nhật. Các bạn nên phóng to trang vẽ để vẽ dễ dàng và chính xác hơn.



Hình 5.2.3. Vẽ súng cho tháp pháo

Phần cuối cùng được vẽ bằng một hình chữ nhật và bốn hình elip. Sau khi vẽ xong một hình elip, các bạn sử dụng công cụ **Tạo bản sao**  để tạo ra ba hình còn lại.



Hình 5.2.4. Hoàn thành vẽ pháo

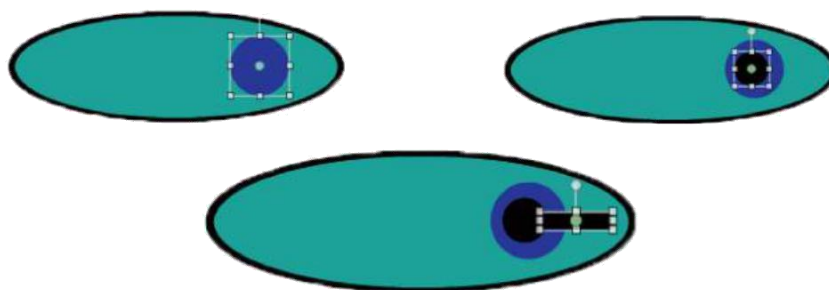
2.2. Vẽ tàu địch

Tàu địch là một đối tượng mới nên ta phải thêm nhân vật bằng cách nháy vào biểu tượng , chuyển hình về dạng Vector. Đầu tiên là vẽ thân tàu bằng một đường viền hình elip. Sau đó đổ màu vào bên trong đường viền này.



Hình 5.2.5. Vẽ thân Tàu địch

Vẽ pháo cho tàu, gồm hai hình tròn là thân pháo và tháp pháo cùng với một hình chữ nhật làm nòng pháo.



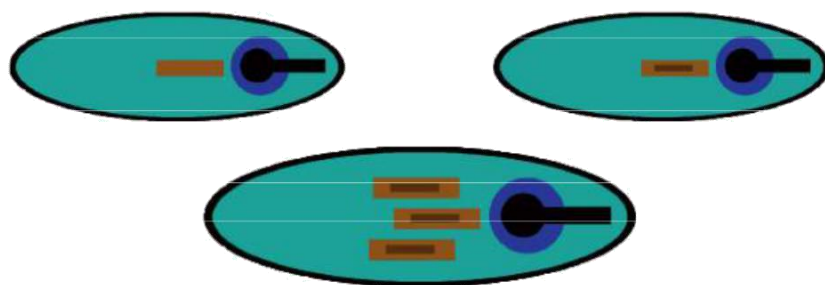
Hình 5.2.6. Vẽ pháo cho tàu

Phần giữa tàu gồm các hình chữ nhật, các bạn nên sử dụng công cụ **Tạo bản sao**



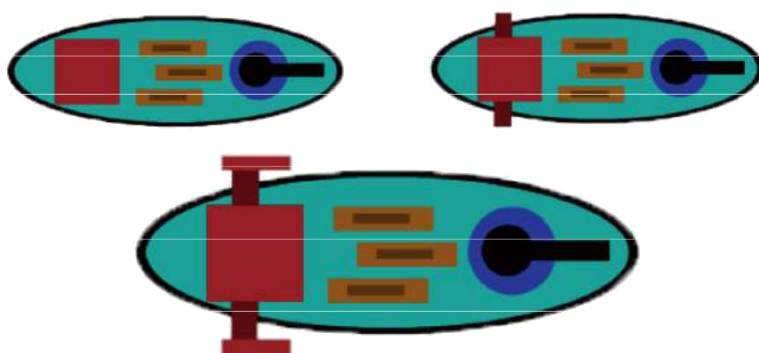
để vẽ nhanh hơn.





Hình 5.2.7. Trang trí phần thân pháo

Phần đuôi tàu được vẽ bằng một hình vuông và bốn hình chữ nhật. Ta nên chỉnh sửa vị trí các hình bằng phím mũi tên trên bàn phím để sắp xếp chính xác hơn.



Hình 5.2.8: Hoàn thành vẽ tàu địch



Tóm tắt lí thuyết và bài tập thực hành

Vẽ Pháo và Tàu Địch

Pháo và Tàu địch

daylaptrinh.net

Các hình vẽ

Pháo

Tàu địch

Công cụ

Đổ màu

Vẽ hình tròn

Vẽ hình vuông

Tạo bản sao

Bài tập

1. Đổi tên cho đối tượng tàu và pháo.
2. Chỉnh lại tâm cho pháo và tàu.
3. Lập trình cho pháo nhảy tới vị trí bắt đầu trong chương trình.
4. Vẽ đối tượng lửa.

BÀI 3 LẬP TRÌNH ĐỐI TƯỢNG PHÁO VÀ TÂM NGẮM

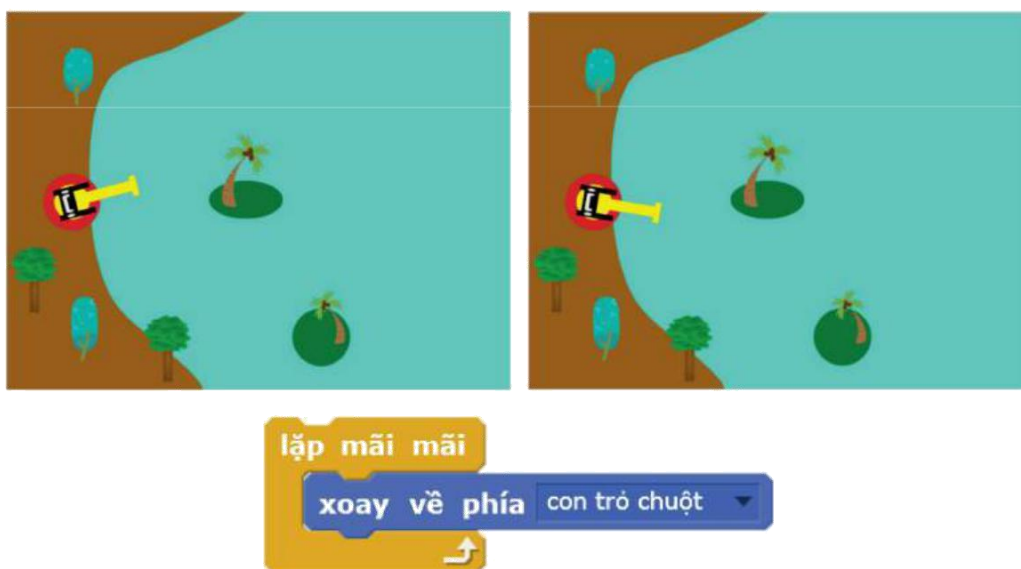
Bài học này chúng ta sẽ lập trình cho đối tượng Pháo và Tâm ngắm. Lưu ý Tâm ngắm ta phải tự vẽ như trong phần bài tập thực hành của Bài 1.

3.1. Lập trình đối tượng Pháo

Trong chương trình, đối tượng Pháo luôn luôn hướng nòng về con trỏ chuột. Sử dụng khối lệnh **xoay về phía con trỏ chuột** để điều khiển nòng pháo quay về phía con trỏ chuột.

Để khối lệnh được thực hiện lặp đi lặp lại, nòng pháo luôn luôn hướng về phía con

trỏ chuột, ta ghép thêm khối lệnh lặp **lặp mãi mãi**.



Hình 5.3.1. Lập trình cho nòng pháo luôn xoay về phía con trỏ chuột

Đoạn khối lệnh hoàn chỉnh cho đối tượng Pháo như sau:



Hình 5.3.2. Đoạn khối lệnh hoàn chỉnh

3.2. Lập trình đối tượng Tâm ngầm

Tâm ngầm luôn luôn di chuyển theo con trỏ chuột. Ta ử dụng khối lệnh

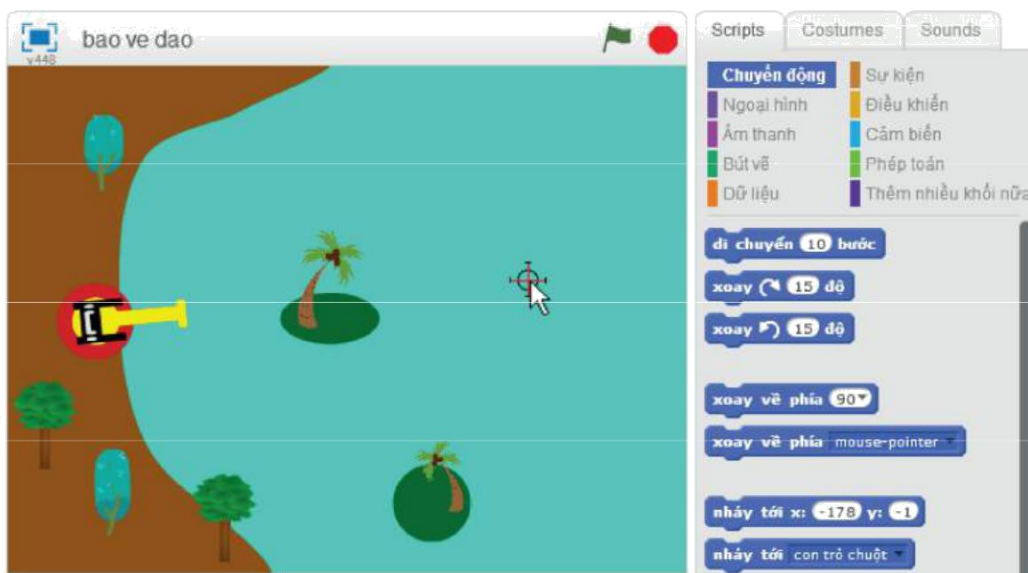
nhảy tới con trỏ chuột để chạy thử. Tâm ngầm di chuyển theo con trỏ chuột.



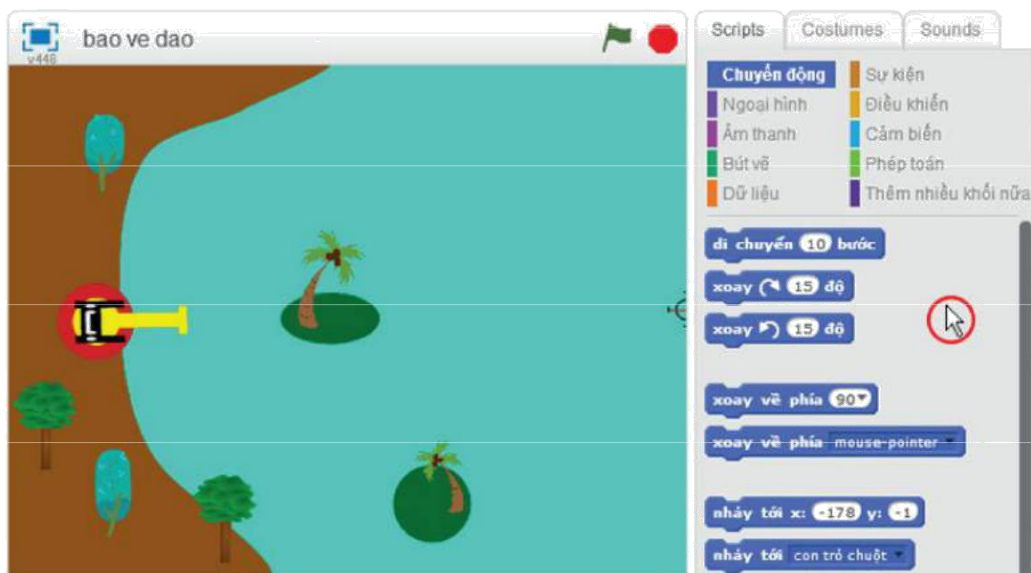
Hình 5.3.3. Chạy thử khối lệnh Nhảy tới con trỏ chuột

Khi chạy khối lệnh, con trỏ chuột đang ở ngoài sân khấu, trong khi các đối tượng chỉ được phép di chuyển bên trong sân khấu. Do đó, Tâm ngầm chỉ di chuyển đến biên rồi

dừng lại. Ghép thêm khối lệnh lặp **lặp mãi mãi** để tâm ngầm luôn di chuyển theo con trỏ chuột.



Khi con trỏ chuột trong sân khấu



Khi con trỏ chuột ra ngoài sân khấu



Hình 5.3.4. Kết quả và đoạn khối lệnh hoàn chỉnh



Tóm tắt lý thuyết và bài tập thực hành

daylaptrinh.net

Lập trình cho Pháo và Tâm ngắm

Pháo và Tâm ngắm

Hình minh họa

Pháo
Tâm ngắm

Đoạn khối lệnh

Đoạn khối lệnh của Tâm ngắm

```

khí nhấn vào
lặp mãi mãi
  nhảy tới con trỏ chuột
            
```

Đoạn khối lệnh của Pháo

```

khí nhấn vào
nhảy tới x: -178 y: -1
lặp mãi mãi
  xoay về phía con trỏ chuột
            
```

Bài tập

- Lập trình cho lửa nhảy tới pháo
- Lập trình cho lửa luôn hướng về phía con trỏ chuột.

BÀI 4

LẬP TRÌNH CHO LỬA

Cũng như vẽ Tâm ngắm, vẽ Lửa sẽ không được hướng dẫn mà được đưa vào bài tập thực hành ở Bài 2. Bài này chúng ta sẽ lập trình cho đối tượng Lửa.

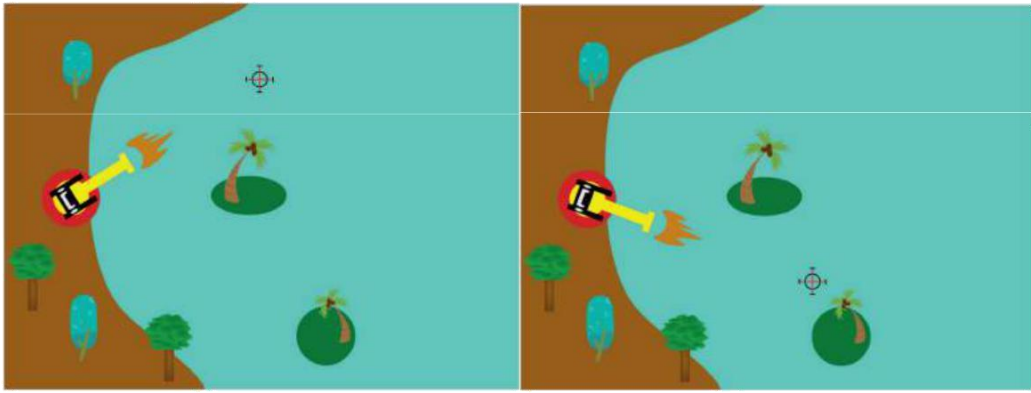
4.1. Lập trình vị trí của Lửa

Đầu tiên, lập trình cho Lửa di chuyển tới Pháo, luôn luôn hướng về phía con trỏ chuột. Đây là lời giải của phần Bài tập thực hành ở Bài 3.



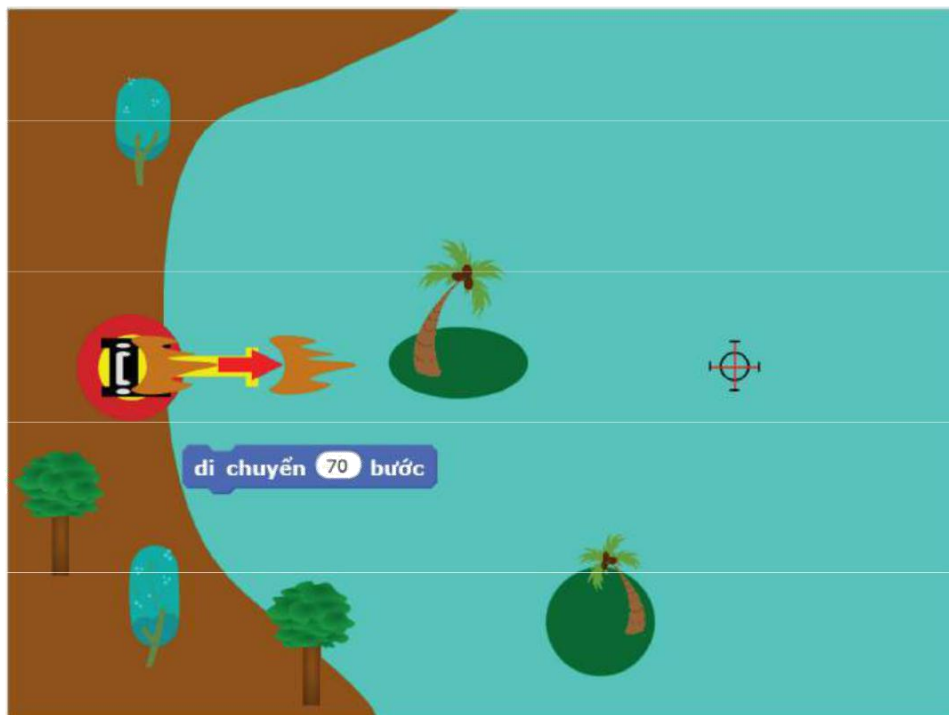
Hình 5.4.1. Lập trình cho Lửa nhảy tới Pháo và hướng về con trỏ chuột

Tiếp theo, lập trình điều khiển Lửa luôn luôn xuất hiện ở đầu nòng pháo chứ không phải ở giữa thân pháo.



Hình 5.4.2. Vị trí của Lửa so với Pháo

Cả hai đối tượng Lửa và Pháo đều cùng quay về phía con trỏ chuột. Lập trình vị trí của Lửa không nằm ở thân Pháo mà ở gần đầu nòng Pháo bằng khối lệnh **di chuyển bước**, số bước cần di chuyển tùy thuộc vào độ dài của nòng pháo.



Hình 5.4.3. Di chuyển Lửa

Vì hướng của nòng pháo luôn hướng về con trỏ chuột và thay đổi liên tục, vậy nên đoạn khối lệnh điều khiển việc nhảy tới Pháo, hướng về phía con trỏ chuột và di chuyển của Lửa phải được thực hiện liên tục bằng khối lệnh lặp.



Hình 5.4.4. Đoạn khối lệnh điều khiển vị trí của Lửa

Khi chạy chương trình, Lửa luôn di chuyển ra phía nòng pháo, nếu i chuyển chuột, nòng pháo và Lửa luôn hướng về phía con trỏ chuột.

4.2. Lập trình đối tượng Lửa chỉ hiện khi nháy chuột

Trong chương trình, khi chúng ta nháy chuột để bắn, Lửa sẽ xuất hiện trong một khoảng thời gian rất ngắn rồi lại biến mất. Để Lửa biến mất, chúng ta sử dụng khối lệnh **ẩn**, khi muốn hiện trở lại, sử dụng khối lệnh **hiện**.



Hình 5.4.5. Khối lệnh ẩn và hiện

Khối lệnh **chuột đã nhấn?** trong nhóm **Cảm biến** sẽ giúp chúng ta kiểm tra xem chuột đã nháy hay chưa. Khối lệnh này trả về cho chúng ta giá trị **false** (sai) nếu chưa nháy chuột, giá trị **true** (đúng) nếu nháy chuột.

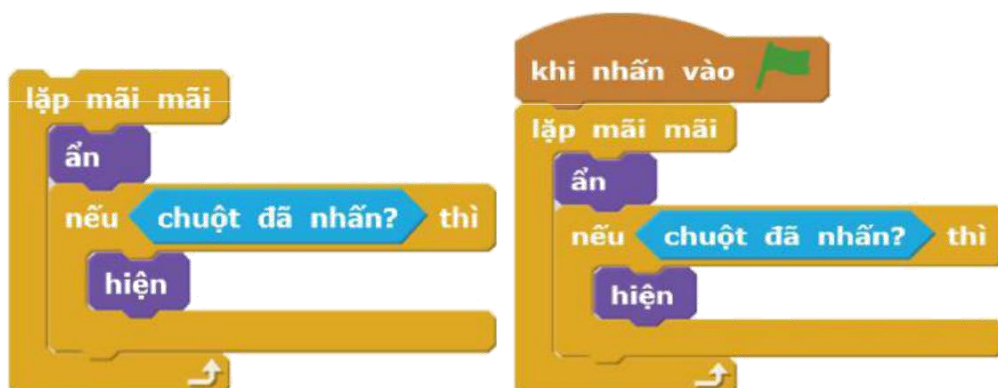
Chúng ta lập trình cho đối tượng Lửa bằng cách sử dụng khối lệnh **Nếu – thì**

 xác định thời điểm ẩn, hiện của đối tượng. Ở trạng thái ban đầu, Lửa được ẩn đi, nếu nhấp chuột thì hiện Lửa lên.



Hình 5.4.6. Đoạn khối lệnh kiểm tra

Ghép thêm khối lệnh **Lặp** để công việc kiểm tra xem có nhấp chuột hay không được thực hiện trong suốt chương trình.



Hình 5.4.7. Đoạn khối lệnh điều khiển sự ẩn, hiện của Lửa



Tóm tắt lí thuyết và bài tập thực hành

Lập trình cho Lửa

Vị trí của Lửa với con trỏ chuột

daylaptrinh.net

Đối tượng

Lửa xuất hiện trước nòng pháo

Đoạn khối lệnh

khí nhấn vào

lặp mỗi mili

nhảy tới Pháo

xoay về phía con trỏ chuột

di chuyển 10 bước

khí nhấn vào

lặp mỗi mili

ẩn

nếu chuột đã nhấn thì

hiện

Đoạn khối lệnh vị trí của Lửa

Đoạn khối lệnh ẩn, hiện của Lửa

Bài tập

- Lập trình để đối tượng Lửa xuất hiện gần đầu nòng pháo hơn.
- Lập trình để khi nhấn và giữ chuột, thì Lửa sẽ liên tục ẩn và hiện, chứ không luôn luôn hiện. Gợi ý: thêm khối lệnh đợi



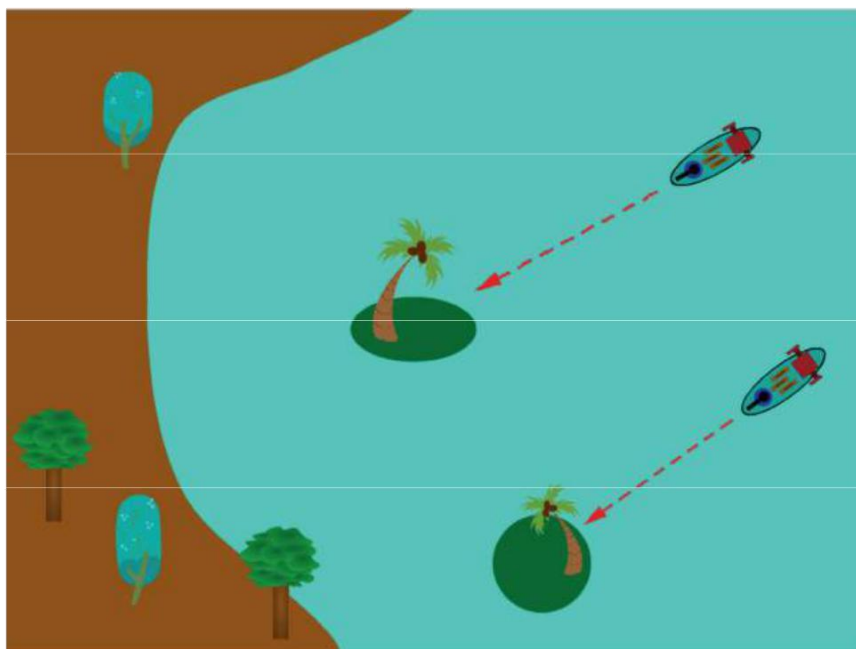
BÀI 5

LẬP TRÌNH CHO TÀU ĐỊCH

Lập trình đối tượng Tàu địch tương đối phức tạp nên được chia ra làm nhiều bài học để các bạn dễ dàng nắm bắt hơn.

5.1. Vị trí của tàu khi hướng về Đảo 1

Tàu địch xuất hiện ở vị trí ngẫu nhiên từ phía bên phải hoặc phía trên của sân khấu. Trong chương trình, tàu xuất hiện ở phía trên sân khấu sẽ tiến đến Đảo 1, còn Tàu xuất hiện ở bên phải sân khấu sẽ tiến đến Đảo 2.



Hình 5.5.1. Vị trí và hướng của Tàu địch

Ta lập trình cho đối tượng Tàu địch. Việc Tàu xuất hiện ở phía trên hay bên phải là ngẫu nhiên. Tạo biến tên là **hướng**, đặt cho biến giá trị ngẫu nhiên 1 hoặc 2
 đặt **hướng** là lấy ngẫu nhiên trong khoảng 1 tới 2. Kiểm tra giá trị của biến **hướng**
hướng = 1, nếu **hướng** bằng 1 thì tàu sẽ xuất hiện ở bên trên và hướng về Đảo 1, còn nếu không thì Tàu sẽ xuất hiện ở bên phải và hướng về Đảo 2.



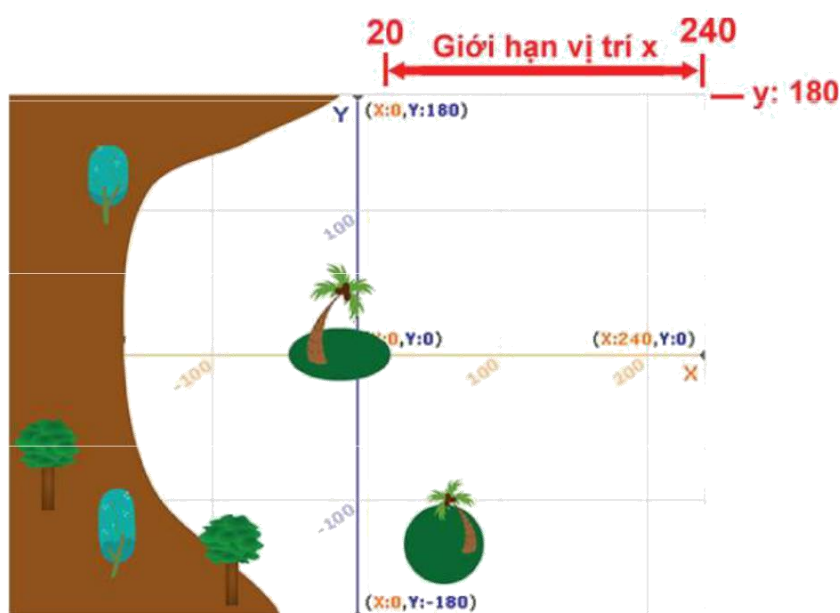
Sử dụng khối lệnh để kiểm tra tàu địch sẽ đi về hướng Đảo 1 hay Đảo 2. Khối lệnh được mô tả như trong Hình 5.5.2.



Hình 5.5.2. Khối lệnh Nếu – thì – còn không thì

Khối lệnh sẽ kiểm tra nếu biến **hướng** bằng 1 thì sẽ thực hiện công việc 1 **làm công việc 1**, còn nếu không (**hướng** khác 1) thì sẽ thực hiện công việc 2 **làm công việc 2**. Trong trường hợp này, nếu **hướng** bằng 1 thì Tàu đích xoay về phía Đảo 1 **xoay về phía Đảo 1** và xuất hiện tại vị trí bên trên.

Khối lệnh xoay sẽ điều khiển Tàu xoay về phía nhân vật khác, trong trường hợp này là Đảo 1. Vị trí bên trên có $yy = 180$ và xx thay đổi được từ 20 đến 240, lấy ngẫu nhiên giá trị trong khoảng 20 đến 240 và ghép vào giá trị xx trong khối lệnh nhảy tới



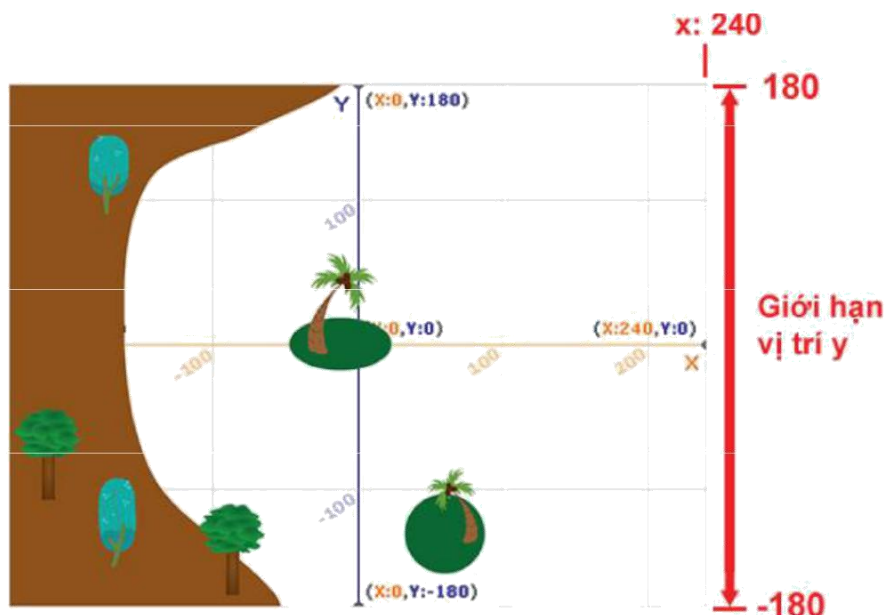
Hình 5.5.3. Tàu đích xuất hiện ở vị trí biên phía trên sân khấu



Hình 5.5.4. Lập trình cho Tàu địch trong trường hợp là tàu tiến về Đảo 1

5.2. Vị trí của tàu khi hướng về Đảo 2

Tương tự như vậy, nếu *hướng* khác 1 thì Tàu địch nhảy tới vị trí bất kì ở biên bên phải và xoay về phía Đảo 2 **xoay về phía Đảo 2**. Biên bên phải có vị trí $xx = 240$ và có yy thay đổi từ 180 đến -180 , vậy ta cần khối lệnh



Hình 5.5.5. Tàu địch xuất hiện ở vị trí biên phải sân khấu



Hình 5.5.6. Đoạn khối lệnh hoàn chỉnh cho sự xuất hiện của Tàu địch



Tóm tắt lí thuyết và bài tập thực hành

Lập trình cho Tàu địch

Hướng và vị trí của Tàu địch

The diagram illustrates the movement area for the enemy boat. It shows a coordinate system with the x-axis ranging from 20 to 240 and the y-axis ranging from -180 to 180. The boat is shown at various positions within this area, including (20, 180), (240, 180), (240, -180), and (20, -180).

daylaptrinh.net

Hướng của Tàu địch

The diagram illustrates the movement area for the enemy boat. It shows a coordinate system with the x-axis ranging from 20 to 240 and the y-axis ranging from -180 to 180. The boat is shown at various positions within this area, including (20, 180), (240, 180), (240, -180), and (20, -180).

Khối lệnh

```

đặt hướng là lấy ngẫu nhiên trong khoảng 1 tới 2
nếu hướng = 1 thì
    nhảy tới x: lấy ngẫu nhiên trong khoảng 20 tới 240 y: 180
    xoay về phía Đảo 1
còn không thì
    nhảy tới x: 240 y: lấy ngẫu nhiên trong khoảng -180 tới 180
    xoay về phía Đảo 2
    
```

Bài tập

- Thêm khối lệnh kích hoạt khi nhảy vào nút lệnh lá cờ xanh cho đoạn khối lệnh.
- Thêm các hình ảnh khác cho tàu ở phần tài liệu vào Khu ngoại hình.



BÀI 6

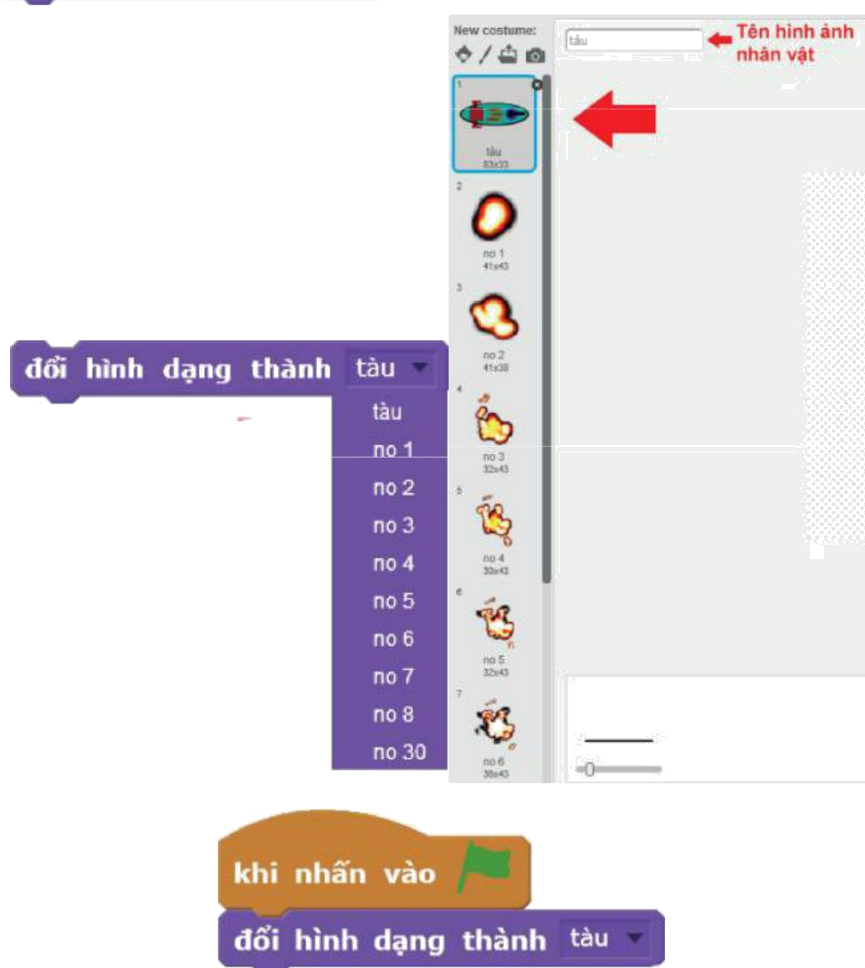
LẬP TRÌNH CHO TÀU ĐỊCH (TIẾP THEO)

Trong chương trình sẽ xuất hiện rất nhiều tàu địch. Trong bài học này các bạn sẽ được hướng dẫn cách tạo bản sao cho Tàu địch và điều khiển những bản sao đó di chuyển tới mục tiêu.

6.1. Lập trình cho bản sao

Ở bài tập thực hành của bài 5, ta đã thêm hình ảnh để mô phỏng hiện tượng nổ cho nhân vật Tàu địch. Những hình ảnh này sẽ được sử dụng khi tàu bị bắn và sẽ được hướng dẫn trong bài sau.

Khi bắt đầu chương trình, đổi lại hình ảnh nhân vật Tàu về trạng thái ban đầu bằng khối lệnh **đổi hình dạng thành tàu**.



Hình 5.6.1. Đặt hình ảnh nhân vật về hình ảnh ban đầu

Sau khi di chuyển đến vị trí thích hợp và hướng về phía một trong hai quần đảo, Tàu địch sẽ tạo ra bản sao của chính nó bằng khối lệnh **tạo bản sao từ myself**. Bản chính sẽ phải lặp lại việc nhảy đến vị trí thích hợp và tạo ra bản sao một cách liên tục, mỗi lần cách nhau 1 giây.



Hình 5.6.2. Tạo bản sao của đối tượng Tàu địch

Lúc này thì bản sao của tàu đã được tạo ra nhưng chưa thực hiện công việc gì cả. Để lập trình cho bản sao di chuyển, ta cần sử dụng khối lệnh **khi tôi khởi đầu là một bản sao**. Với khối lệnh này, khi một bản sao được tạo ra, tất cả các khối lệnh ghép dưới khối lệnh này sẽ được thực hiện cho bản sao đó.

Khi bản sao được tạo, hướng của bản sao cũng chính là hướng của bản chính, trong khi bản chính đã hướng sẵn về phía một trong hai quần đảo, vậy bản sao chỉ cần di chuyển về phía trước bằng khối lệnh **di chuyển 2 bước**.

Bản sao sẽ di chuyển cho đến khi chạm vào Đảo 1 hoặc Đảo 2 thì dừng.

Trong trường hợp này chúng ta sẽ sử dụng khối lệnh lặp



Ta hãy xem xét cách sử dụng khối lệnh lặp này trong Hình 5.6.3.



Hình 5.6.3. Khối lệnh Lặp cho tới khi

Khi chạy đến khối lệnh này, khối lệnh **điều kiện** sẽ được kiểm tra xem là đúng hay sai. Nếu sai thì khối lệnh **làm công việc 1** được thực hiện lặp đi lặp lại, cho đến khi **điều kiện** trở thành đúng thì sẽ dừng vòng lặp và thực hiện **làm công việc 2**.

Bản sao của Tàu địch sẽ sử dụng khối lệnh như sau:



Hình 5.6.4. Đoạn khối lệnh điều khiển bản sao di chuyển

Trong trường hợp này, tàu sẽ lặp lại việc di chuyển cho đến khi **đang chạm vào Đảo 1 ?** hoặc **đang chạm vào Đảo 2 ?** thì dừng lại. Ở đây, khối lệnh **đang chạm vào Đảo 1 ?** sẽ kiểm tra xem đối tượng Tàu địch này có chạm vào đối tượng Đảo 1 hay không, nếu có sẽ trả về **true** (đúng), còn nếu không thì sẽ là giá trị **false** (sai).

Tương tự, khối lệnh **đang chạm vào Đảo 2 ?** sẽ kiểm tra xem Tàu có chạm vào đối tượng Đảo 2 hay không.

Chỉ cần một trong hai khối lệnh kiểm tra có chạm vào đảo hay không trả về giá trị **true** (đúng) thì cả khối lệnh **đang chạm vào Đảo 1 ?** hoặc **đang chạm vào Đảo 2 ?** sẽ mang giá trị đúng. Tức là việc di chuyển sẽ được thực hiện liên tục cho đến khi tàu chạm vào một trong hai đối tượng Đảo 1 hoặc Đảo 2 (hoặc Đảo 1 hoặc Đảo 2) thì dừng lại.

BÀI 7

LẬP TRÌNH CHO TÀU ĐỊCH (TIẾP THEO)

Trong bài học này, các bạn sẽ lập trình để Tàu địch bị bắn trúng sẽ nổ và người chơi được cộng điểm.

7.1. Lập trình cho Tàu địch bị bắn

Người chơi sẽ bắn Tàu địch bằng cách nháy chuột lên tàu. Trong khi bản sao của tàu di chuyển đến Đảo 1 hoặc Đảo 2, nếu Pháo bắn (người chơi nháy chuột), mà tâm ngắm lại chạm bản sao (tàu bị bắn trúng) thì điểm sẽ được cộng, bản sao tàu sẽ nổ và biến mất.

Bản sao của Tàu địch sẽ kiểm tra việc Pháo bắn hay không trong lúc di chuyển về phía quần đảo.



Hình 5.7.1. Kiểm tra xem Pháo có bắn hay không

Khối lệnh 'chuột đã nhấn?' sẽ trả về giá trị **true** (đúng) nếu người chơi nháy chuột. Khi người chơi nháy chuột, (khối lệnh 'chuột đã nhấn?' có giá trị đúng) thì kiểm tra tiếp xem Tàu địch có đang chạm vào Tâm ngắm (Pháo bắn trúng) không.

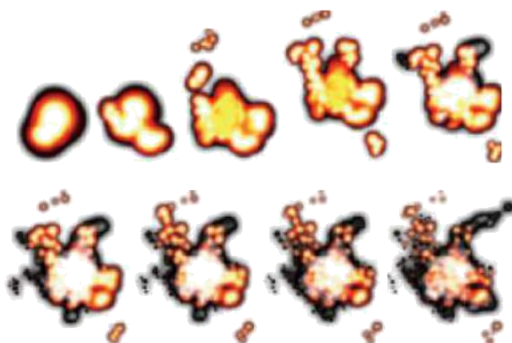


Hình 5.7.2. Kiểm tra xem Pháo có bắn trúng Tàu địch không



Nếu Pháo bắn trúng (người chơi nháy chuột vào Tàu địch) thì điểm sẽ được cộng thêm 1 và tàu sẽ nổ rồi biến mất. Trong phần bài tập thực hành của bài 6, ta đã tạo biến **điểm**, để tăng biến **điểm** lên ta sử dụng khối lệnh **thay đổi điểm** một lượng **1**.

Từ bài học thực hành của bài 5, ta đã thêm 30 hình ảnh vào khu ngoại hình để thể hiện hình ảnh từ khi phát nổ đến khi tro tàn của Tàu địch, ta chỉ cần lập trình để thay đổi lần lượt qua 30 hình ảnh này liên tục là tàu sẽ phát nổ.

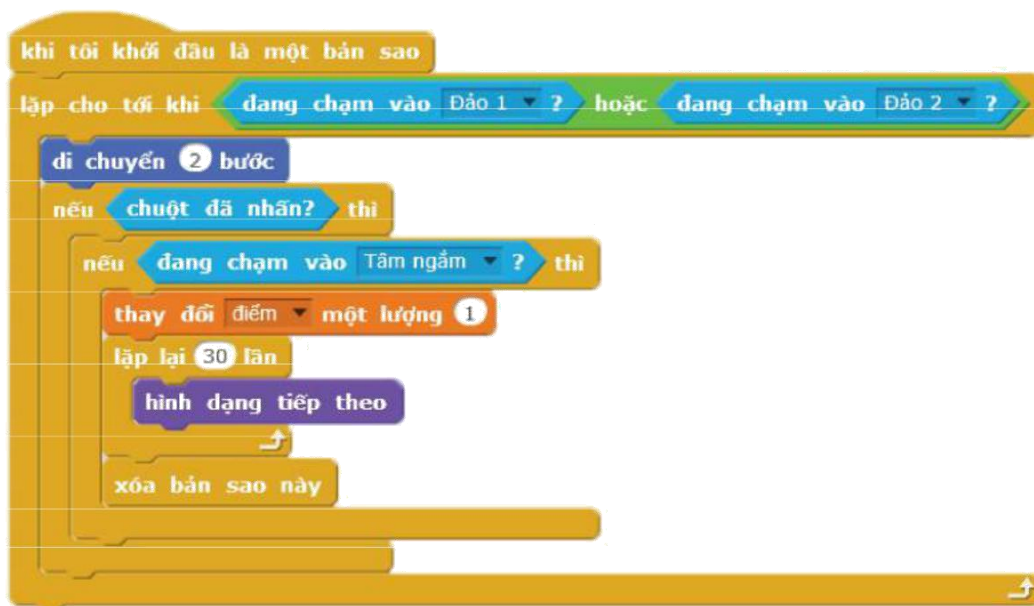


Hình 5.7.3. Một số hình ảnh nhân vật tàu phát nổ



Hình 5.7.4. Lập trình cho tàu phát nổ

Sau khi nổ xong vẫn còn lại hình tro tàn. Chúng ta phải xóa bản sao này đi bằng khối lệnh **xóa bản sao này**. Cuối cùng ghép các khối lệnh này lại với nhau.



Hình 5.7.5. Thêm đoạn khối lệnh cho bản sao của Tàu địch



Tóm tắt lí thuyết và bài tập thực hành

Tàu địch bị bắn

Lập trình cho Tàu địch phát nổ

daylaptrinh.net

Hình ảnh nhân vật Tàu địch phát nổ

Khối lệnh

Bài tập

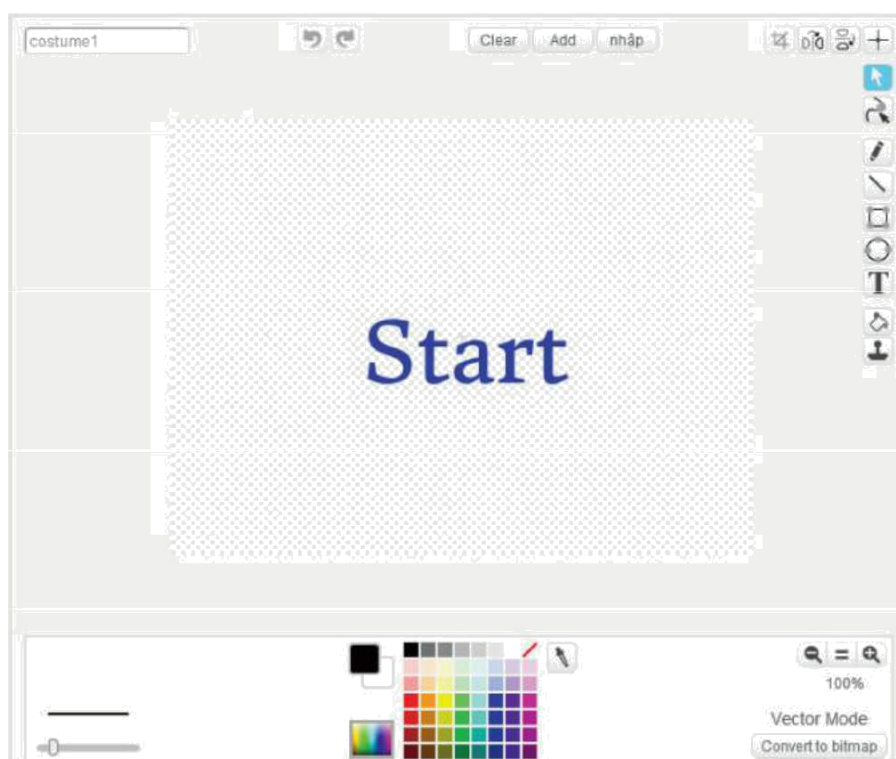
1. Lập trình để khi bản sao của Tàu địch chạm vào Đảo 1 hoặc Đảo 2 thì tất cả chương trình sẽ dừng lại.
2. Thêm âm thanh cho khi Tàu địch nổ và lập trình để khi Tàu địch nổ âm thanh cũng được chạy.

BÀI 8**KẾT THÚC CHƯƠNG TRÌNH**

Đến đây chương trình đã gần được hoàn thành. Bài cuối cùng này sẽ hoàn thiện thêm để chương trình có bắt đầu và kết thúc rõ ràng.

8.1. Thêm và lập trình đối tượng chữ Start

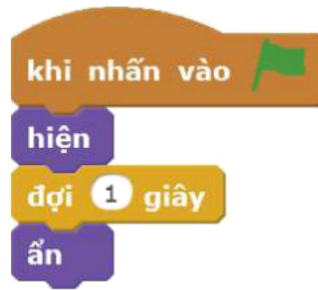
Khi nhấp vào nút lệnh lá cờ xanh bắt đầu chương trình, chữ Start sẽ hiện lên giữa sân khấu, báo hiệu trò chơi được bắt đầu. Ta cần thiết kế đối tượng này bằng cách nhấp vào biểu tượng  ở Khung  nhân vật rồi chuyển đến Khu ngoại hình, huyển hình ảnh về chế độ Vector. Chúng ta sẽ sử dụng công cụ  để thêm chữ Start. Ta có thể chỉnh kích thước, màu sắc, phong chữ tùy ý.



Hình 5.8.1. Thêm đối tượng chữ Start

Khi bắt đầu, đối tượng chữ Start này sẽ hiện lên trong vòng 1 giây rồi biến mất. Chúng ta sẽ sử dụng lần lượt các khối lệnh ,  và .

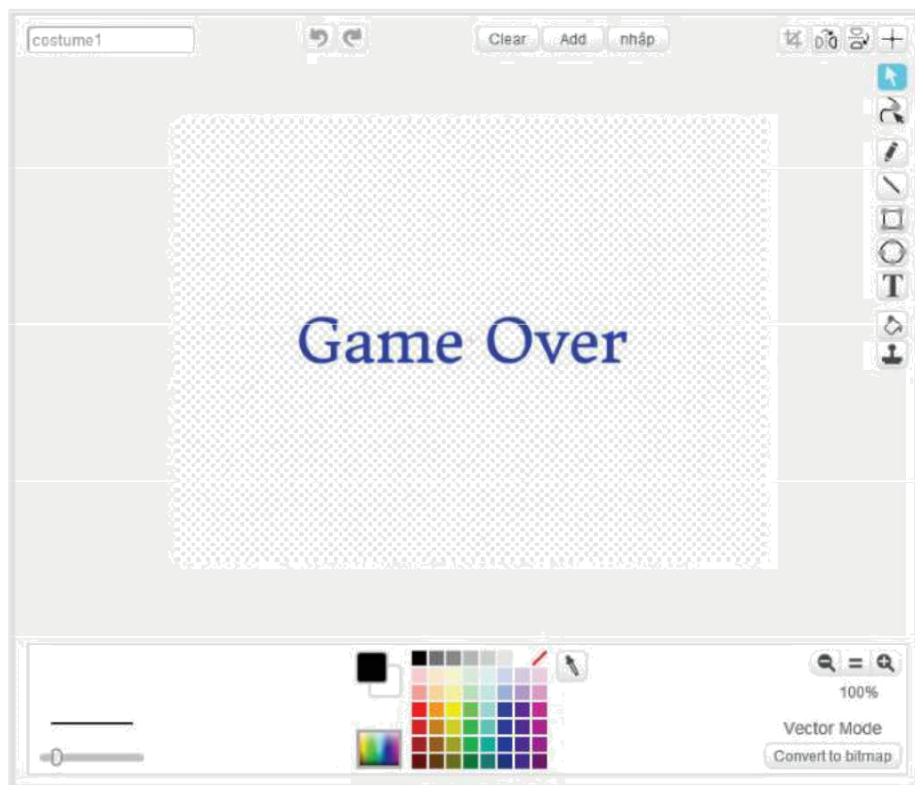




Hình 5.8.2. Lập trình cho đối tượng Start

8.2. Thêm và lập trình đối tượng chữ Game Over

Việc thêm mới và thiết kế đối tượng chữ Game Over cũng làm tương tự như đối tượng chữ Start.



Hình 5.8.3. Thêm đối tượng chữ Game Over.

Tính từ lúc bắt đầu chạy, trong suốt chương trình đối tượng chữ Game Over luôn được ẩn đi.



Hình 5.8.4. Đoạn khối lệnh ẩn đối tượng Game Over

Đến khi kết thúc chương trình chữ Game Over sẽ hiện lên. Để đối tượng chữ Game Over này biết được thời điểm kết thúc chúng ta cần lập trình cho bản sao của tàu thông báo kết thúc tới tất cả các đối tượng khác trong chương trình khi chạm vào một trong hai đảo.

Để làm thông báo, ta sẽ sử dụng khối lệnh **thông báo** **message1** và tạo một thông báo mới bằng cách nháy vào mũi tên trong khối lệnh và chọn *new message*.



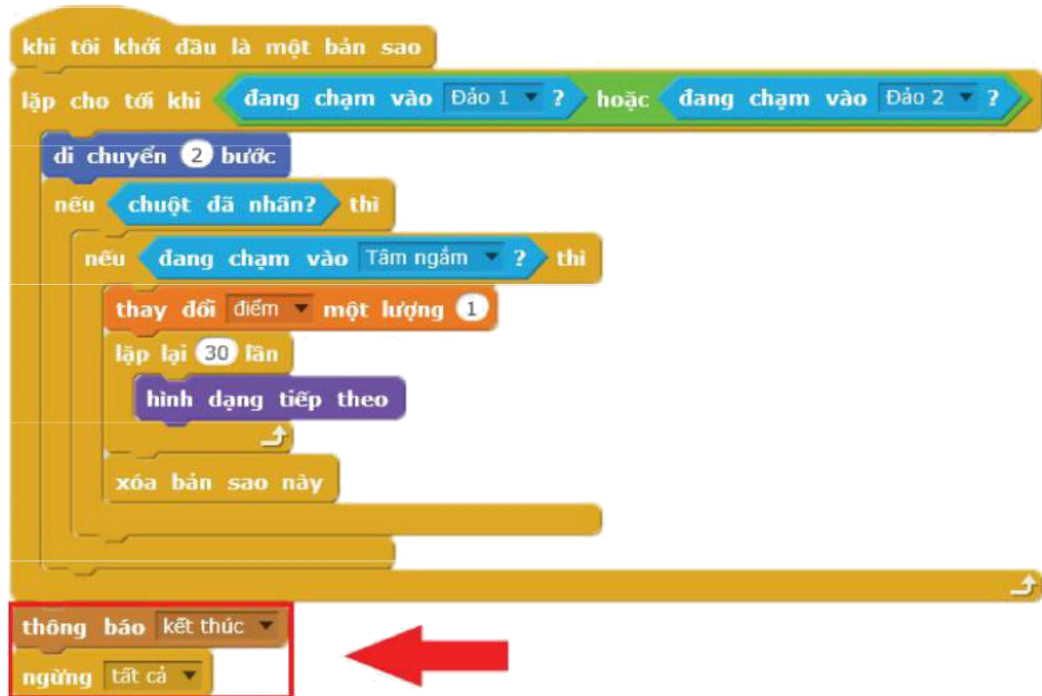
Hình 5.8.5. Tạo thông báo

Lúc này, một cửa sổ sẽ hiện ra để bạn đặt tên cho thông báo. Sau khi đặt tên xong các bạn nhấn nút **OK**.



Hình 5.8.6. Đặt tên cho thông báo

Các bạn ghép khối lệnh **thông báo** **kết thúc** xuống dưới khối lệnh lập trình cho bản sao của đối tượng Tàu địch. Ở vị trí này, khi điều kiện **đang chạm vào Đảo 1** ? hoặc **đang chạm vào Đảo 2** ? đúng thì khối lệnh thông báo này sẽ được chạy. Sau khi thông báo xong chương trình sẽ ngừng bằng khối lệnh **ngừng** **tất cả**. Khối lệnh này cũng là đáp án câu 1 phần bài tập thực hành của bài 7.



Hình 5.8.7. Đoạn khối lệnh hoàn chỉnh của bản sao đối tượng Tàu địch

Bây giờ tất cả các đối tượng ng khác đều có thể nhận được thông báo kết thúc bằng khối lệ h **khi tôi nhận được thông báo kết thúc**. Ta sẽ phải lập trình cho đối tượng chữ Game Over để khi nhận được thông báo kết thúc này sẽ hiện lên trên sân khấu.



Hình 5.8.8. Đoạn khối lệnh nhận thông báo

Vậy là chúng ta đã lập trình xong cho chương trình “Bảo vệ đảo”. Các bạn cần hoàn thành đầy đủ bài tập ở cuối mỗi bài để chương trình hoàn thiện hơn. Các bạn cũng có thể tự sáng tạo thêm các loại tàu khác, cũng như tăng số lượng và tốc độ của tàu để trò chơi thú vị hơn.



Tóm tắt lí thuyết và bài tập thực hành

Kết thúc chương trình

Lập trình cho chữ Start và Game Over

daylaptrinh.net

Đối tượng Start

khi nhấn vào

hiện

đợi 1 giây

ẩn

Đối tượng Game Over

khi nhấn vào

ẩn

khi tôi nhận được thông báo kết thúc

hiện

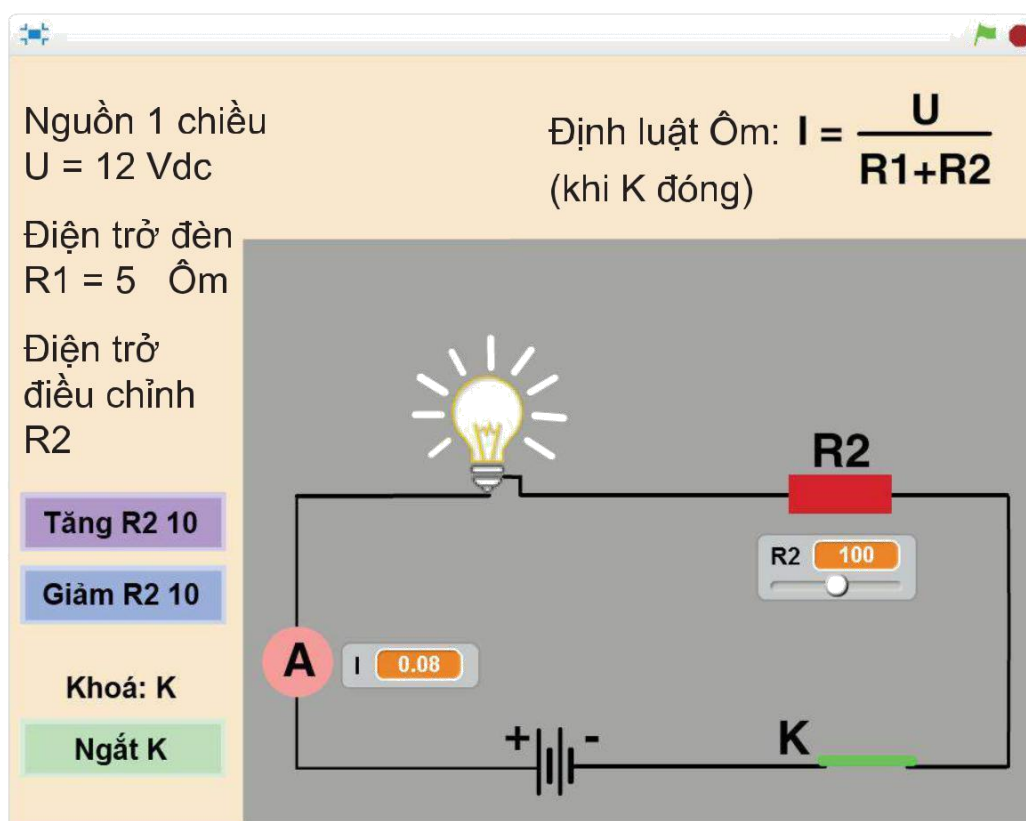
Bài tập

- Lập trình để đối tượng Start và Game Over nhảy tới vị trí nhất định
- Lập trình để bản chính của Tàu địch ẩn đi

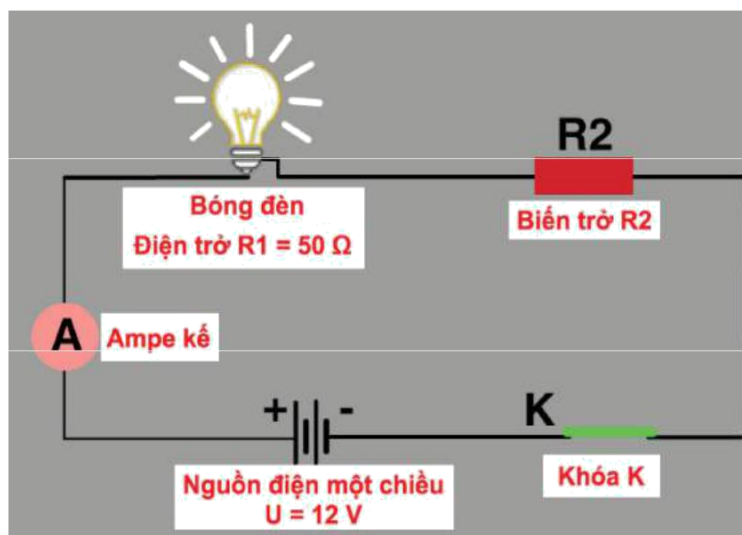


PHỤ LỤC**Mô phỏng mạch điện**

Đây là phần hướng dẫn ngắn gọn về chương trình Mô phỏng mạch điện. Chương trình được ứng dụng kiến thức về định luật Ôm trong môn Vật lí để mô phỏng hoạt động của một mạch điện đơn giản.

Tổng quan chương trình

Trong chương trình, mạch điện gồm bóng đèn có điện trở R1 = 50 Ω, nguồn điện, khoá K, biến trở R2, ampe kế được mô tả ở Hình 6.1.1.



Hình 6.1.1. Mô tả mạch điện

Khi khoá K đóng, trong mạch điện xuất hiện dòng điện với cường độ I được tính bằng công thức định luật Ôm:

$$I = \frac{U}{R_1 + R_2}$$

Trong công thức trên, U là hiệu điện thế giữa hai cực của nguồn điện một chiều và có giá trị cố định bằng 12 vôn, R_1 là điện trở của bóng đèn, R_2 là điện trở của biến trở.

Chương trình gồm các đối tượng:

Khoá K: Đóng vai trò như một công tắc của mạch điện, điều khiển mạch đóng hoặc ngắt.

Công tắc: Điều khiển khoá K, gồm hai hình ảnh nhân vật tương ứng với hai trạng thái đóng hoặc ngắt của khoá K.

Bóng đèn: Gồm nhiều hình ảnh đối tượng tương ứng với các trạng thái khác nhau của đèn.

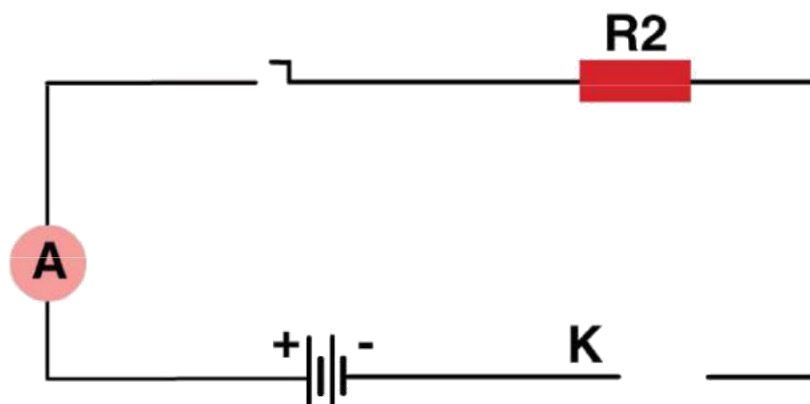
Tăng R2 10: Tăng R2: Điều khiển tăng điện trở của biến trở 10 Ω.

Giảm R2 10: Giảm R2: Điều khiển giảm điện trở của biến trở 10 Ω.

Thiết kế các đối tượng

1. Vẽ mạch điện

Mạch điện được thiết kế như Hình 6.1.2:



Hình 6.1.2. Mạch điện

Trong mạch điện, đường dây điện được vẽ bằng công cụ , điện trở được vẽ bằng công cụ , Ampe kế được vẽ bằng công cụ , các số và chữ cái được thêm bằng công cụ . Sau đó sử dụng một hình chữ nhật tối màu để làm nền. Chúng ta nhớ đặt hình chữ nhật này ra sau bằng công cụ  để mạch điện xuất hiện lên phía trên.

2. Vẽ các đối tượng

Các đối tượng Tăng R2, Giảm R2, Công tắc được vẽ bằng công cụ  và .



Hình 6.1.3. Hình ảnh các đối tượng

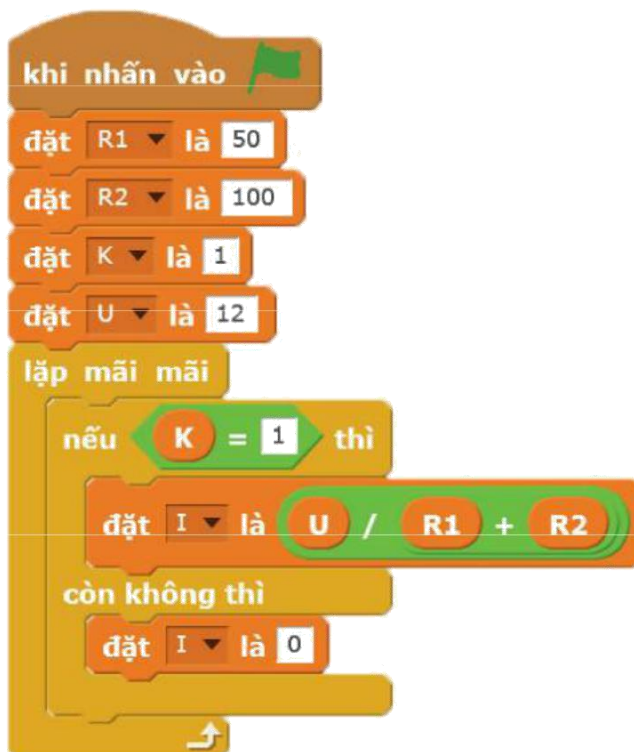
Đối tượng Bóng đèn và các hình ảnh đối tượng được đưa vào trong tài liệu của khoá học.

Lập trình

Các bạn tạo năm biến với lựa chọn *For all sprites*. Các biến có tên và mô tả chức năng như sau:

-  **I**: Chứa giá trị cường độ dòng điện.
-  **K**: Chứa giá trị trạng thái khoá K đóng hoặc ngắt.
-  **R1**: Chứa giá trị điện trở của đèn.
-  **R2**: Chứa giá trị điện trở của biến trở R2.
-  **U**: Chứa giá trị hiệu điện thế giữa hai cực nguồn điện một chiều.

Khi bắt đầu chương trình, Sân khấu khởi tạo giá trị cho các biến. Khi khoá K ngắt, cường độ I có giá trị bằng 0. Khi khoá K mở, I được tính theo công thức định luật Ôm đã được nhắc tới ở trên.



Hình 6.1.4. Khởi tạo giá trị các biến và tính cường độ dòng điện

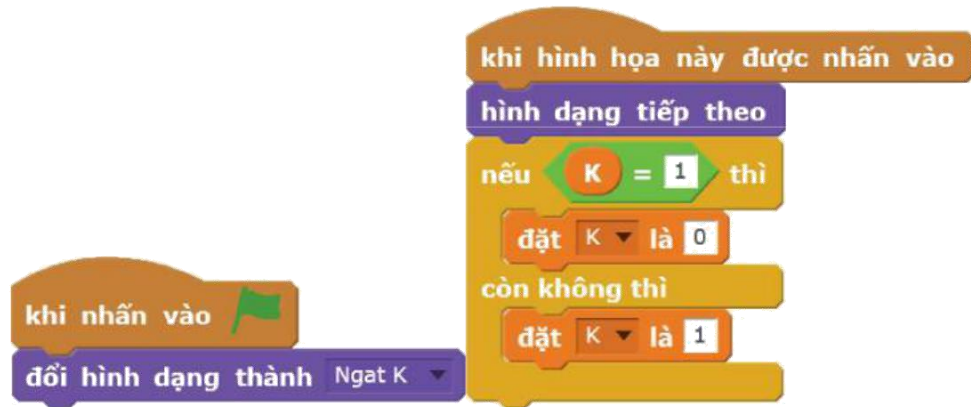
Biến **I** và biến **R2** được hiện trên sân khấu.

Đối tượng khoá K có hai trạng thái đóng hoặc ngắt, phụ thuộc vào biến **K**. Biến chứa giá trị 1 tương ứng với trạng thái đóng, giá trị 0 tương ứng trạng thái ngắt.



Hình 6.1.5. Hai trạng thái đóng và ngắt của khoá K

Khi bắt đầu chương trình, khoá K ở trạng thái đóng, đối tượng công tắc sử dụng biến **K** để điều khiển khoá K.



Hình 6.1.6. Đối tượng công tắc

Hai đối tượng Tăng R2 và Giảm R2 điều chỉnh điện trở R2.

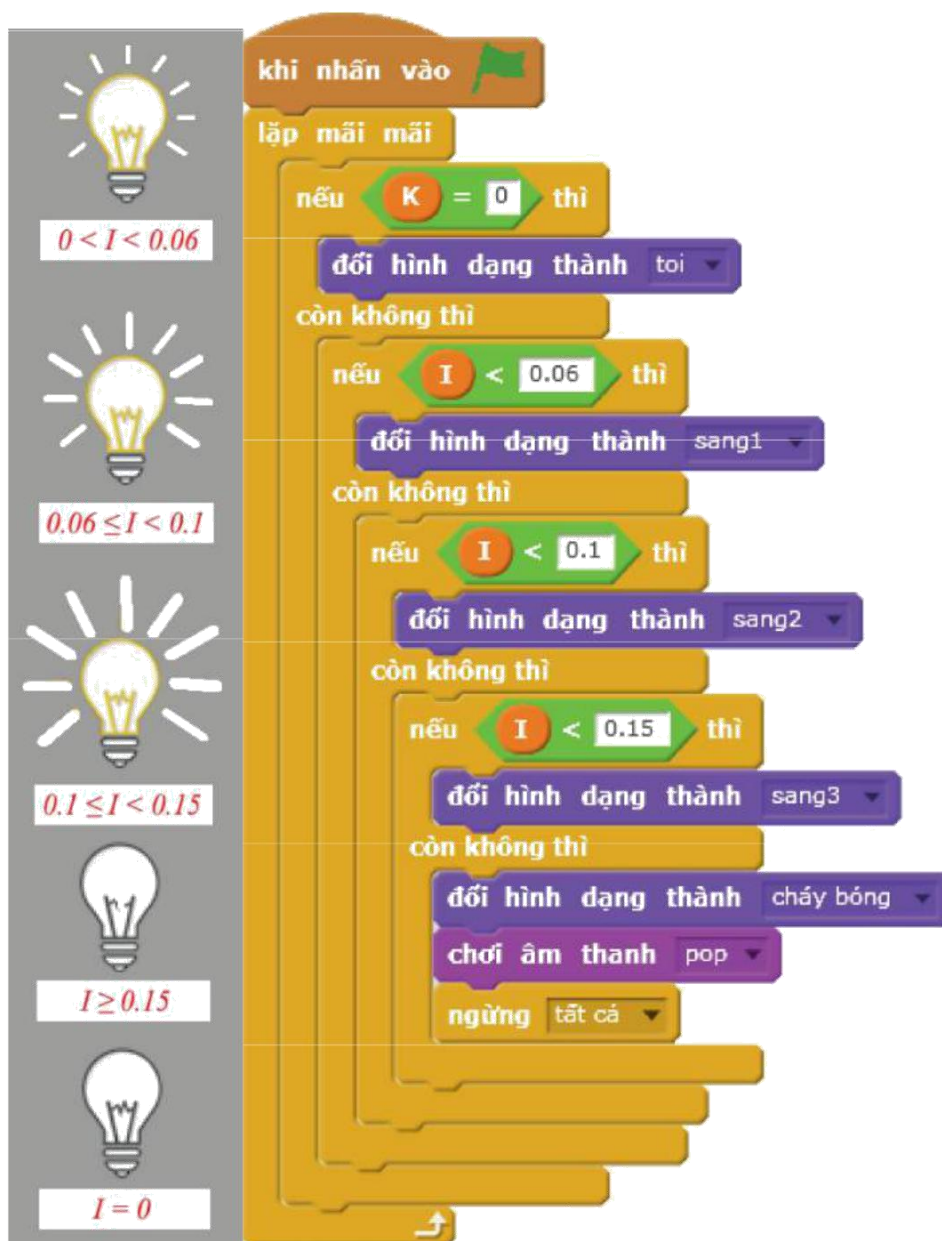


Hình 6.1.7. Tăng R2 10Ω



Hình 6.1.8. Giảm R2 10Ω

Bóng đèn có các hình ảnh là các trạng thái khác nhau, tương ứng với các giá trị của cường độ dòng điện.



Hình 6.1.9. Các trạng thái tương ứng với các giá trị cường độ dòng điện

Sau khi hoàn thành, các bạn có thể phát triển thêm chương trình bằng cách thêm chức năng thay đổi hiệu điện thế U .