

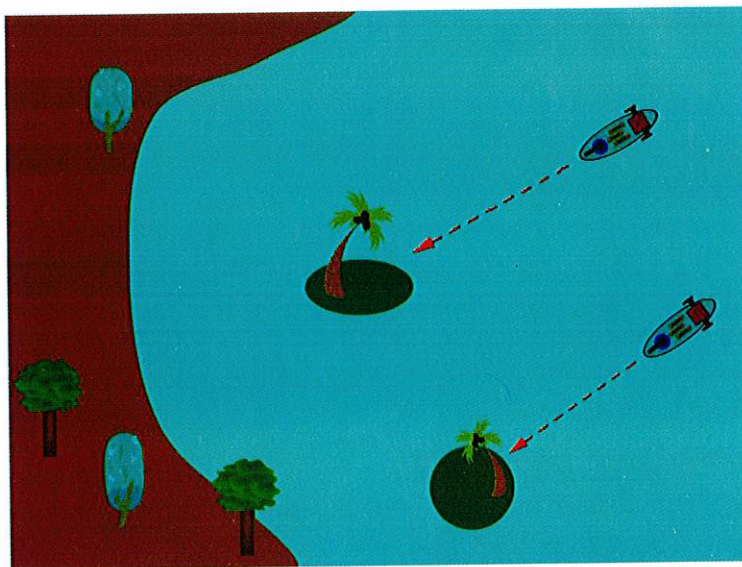
BÀI 5

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** thành lấy ngẫu nhiên từ 1 đến 2. Kiểm tra giá trị của biến **hướng** **hướng** = 1

nếu **hướng** bằng 1, 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.

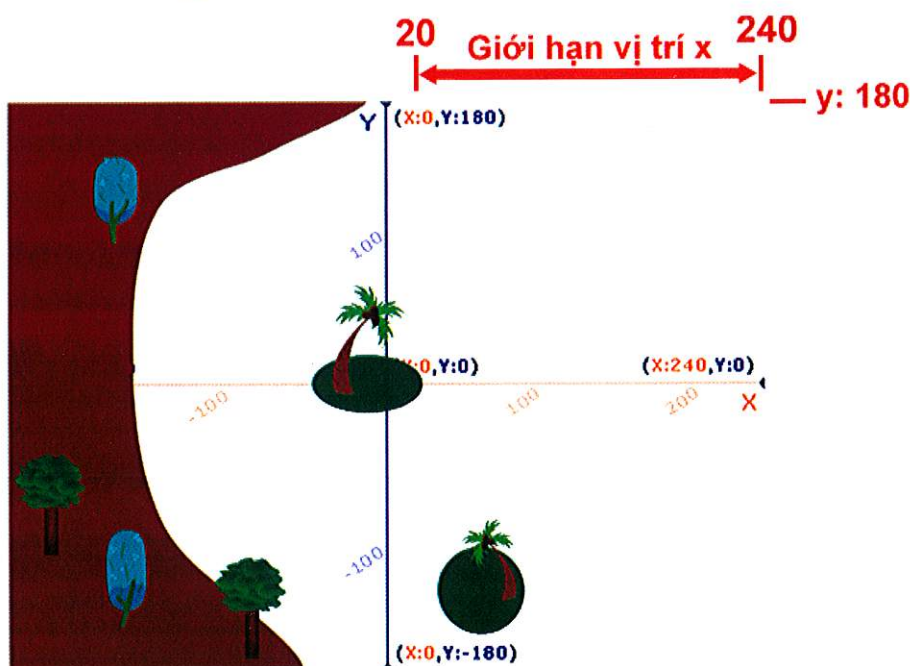




Hình 5.5.2. Khối lệnh Nếu – thì – nếu không thì

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 dịch hướng về vị trí Đảo 1 **hướng về phía đối tượng Đảo 1** và xuất hiện tại vị trí bên trên.

Khối lệnh **hướng về phía đối tượng** sẽ điều khiển Tàu xoay về phía nhân vật, đối tượng khác, trong trường hợp này là Đảo 1. Vị trí phía trên có $y = 180$ và x thay đổi được từ 20 đến 240, ta lấy ngẫu nhiên giá trị trong khoảng 20 đến 240 và ghép vào giá trị x trong khối lệnh đi tới **đi tới điểm x: lấy ngẫu nhiên từ 20 đến 240 y: 180**.



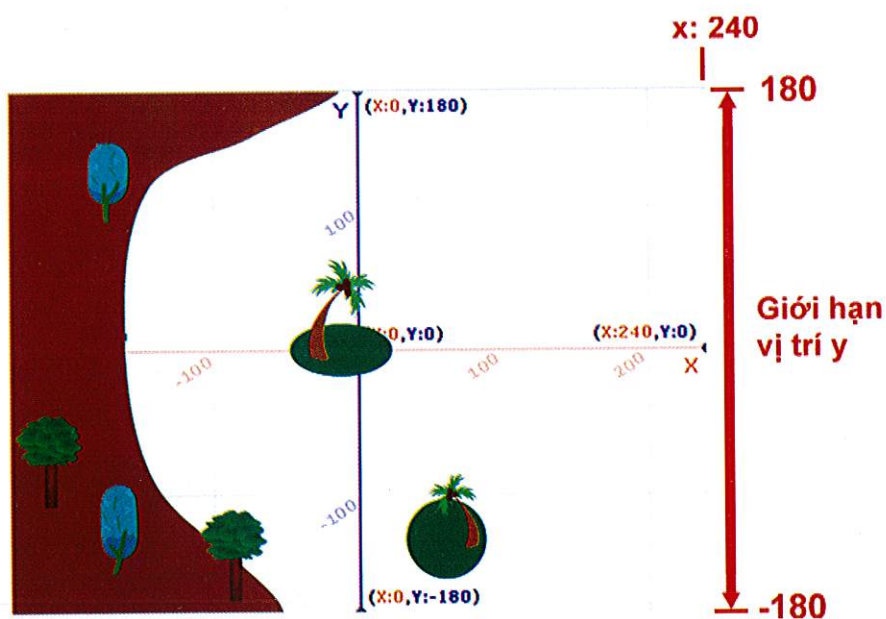
Hình 5.5.3. Tàu dịch xuất hiện ở vị trí biên phía trên sân khấu



Hình 5.5.4. Lập trình cho Tàu dịch 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 dịch đi tới vị trí bất kỳ ở biên bên phải và xoay về phía Đảo 2. Biên bên phải có vị trí $x = 240$ và vị trí y thay đổi ngẫu nhiên từ 180 đến -180 .



Hình 5.5.5. Tàu dịch 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 lý thuyết và bài tập thực hành

Lập trình cho Tàu địch

Hướng và vị trí của Tàu địch

vietstem.com

Hướng của Tàu địch

Khối lệnh

Bài tập

1. Thêm khối lệnh kích hoạt khi nhấn vào nút lệnh lá cờ xanh cho đoạn khối lệnh.
2. Thêm các hình ảnh khác cho tàu ở phần tài liệu vào Khu Trang phục.



BÀI 6

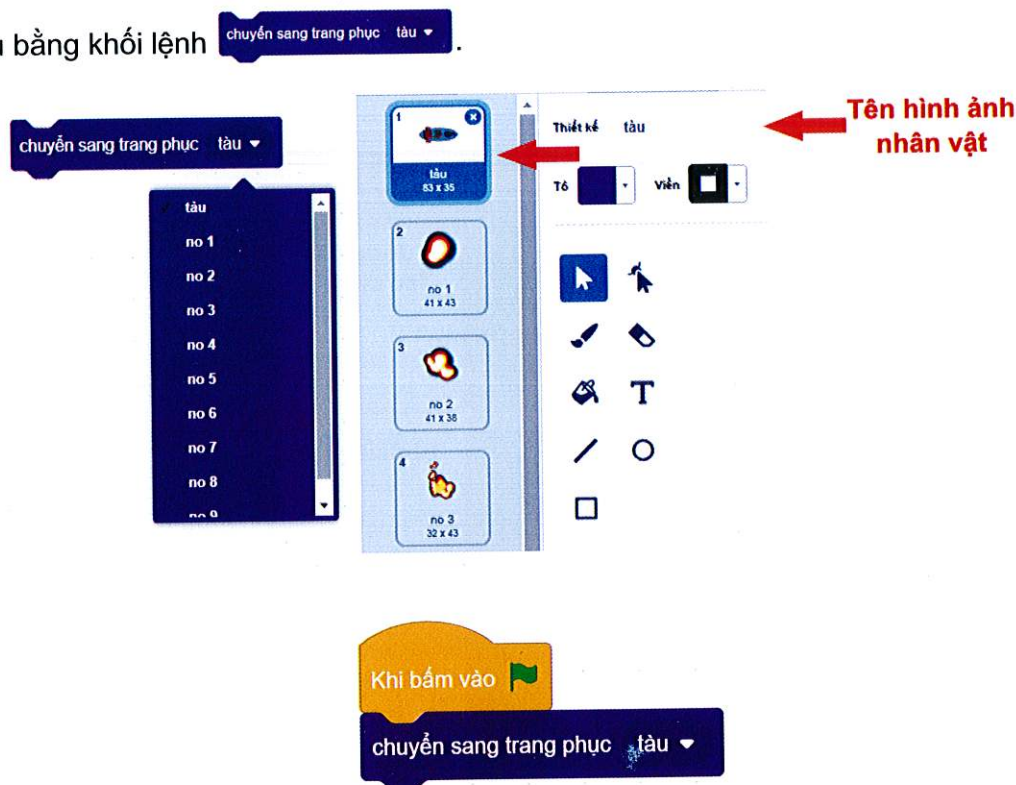
LẬP TRÌNH CHO TÀU ĐỊCH (TIẾP THEO)

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.

Lập trình bản sao Tàu di chuyển

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

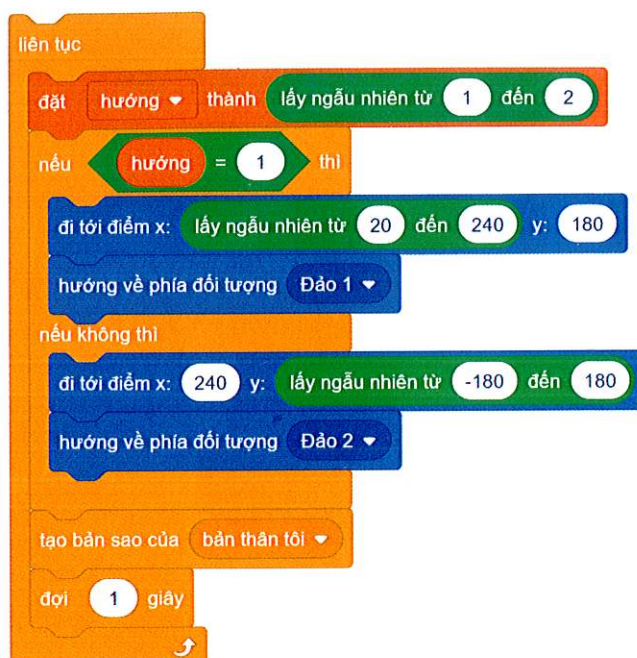
Khi bắt đầu chương trình, đổi lại hình ảnh đối tượng Tàu địch về trạng thái ban đầu bằng khối lệnh



Hình 5.6.1. Đặt hình ảnh đối tượng 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 của** **bản thân tôi**. Bản chính sẽ phải lặp lại việc đi tới 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 bản sao của tàu đã được tạo ra nhưng chưa thực hiện công việc gì. Để lập trình cho bản sao di chuyển, ta cần sử dụng khối lệnh **khởi tạo bản sao**. Với khối lệnh này, khi một bản sao được tạo ra, tất cả các 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**.

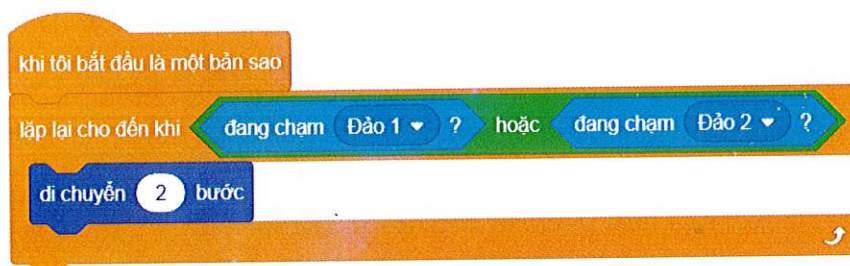
Bản sao sẽ di chuyển cho đến khi chạm vào Đảo 1 hoặc Đảo 2 thì dừng. Trong trường hợp này chúng ta sẽ sử dụng khối lệnh **lặp lại cho đến khi**.

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



Hình 5.6.3. Minh họa khối lệnh Lặp cho đến khi

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

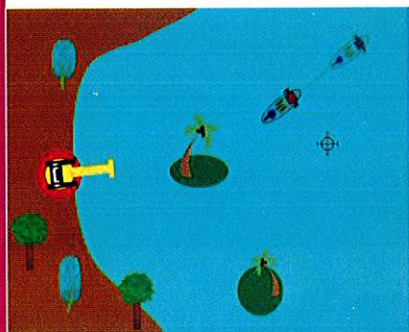
Tương tự, khối lệnh **đang chạm Đảo 2 ?** sẽ kiểm tra xem Tàu có chạm vào Đảo 2 hay không.

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 Đảo 1 ?** hoặc **đang chạm 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.

Tóm tắt lý thuyết và bài tập thực hành

Lập trình cho bản sao

Bản sao di chuyển



vietstem.com

Tạo bản sao

tạo bản sao từ myself

Bản sao di chuyển

khi tôi bắt đầu là một bản sao là sao
lập lại cho đến khi đang chạm Đảo 1 hoặc đang chạm Đảo 2
di chuyển 2 bước

Bài tập

1. Tạo biến điểm.
2. Lập trình để khi nhấn vào nút lệnh lá cờ xanh, biến điểm sẽ nhận giá trị là 0.
3. Lập trình để bản sao di chuyển chậm hơn.

