



# ALICE

*Say mê và  
sáng tạo*



**Giáo viên hướng dẫn**

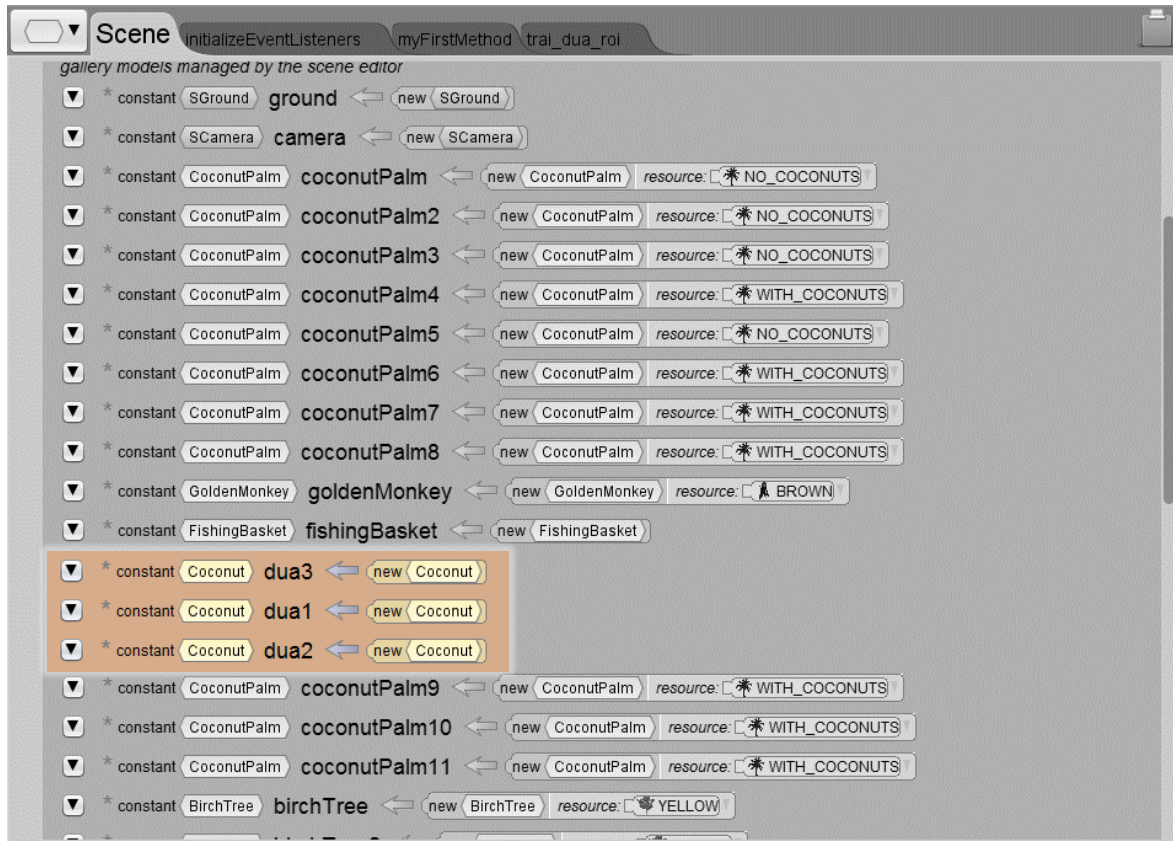
❖ Trần Phúc Minh Khoa

❖ Nguyễn Võ Phi Long

## Sử dụng lập để lập trình trò chơi khi Hùng dừa

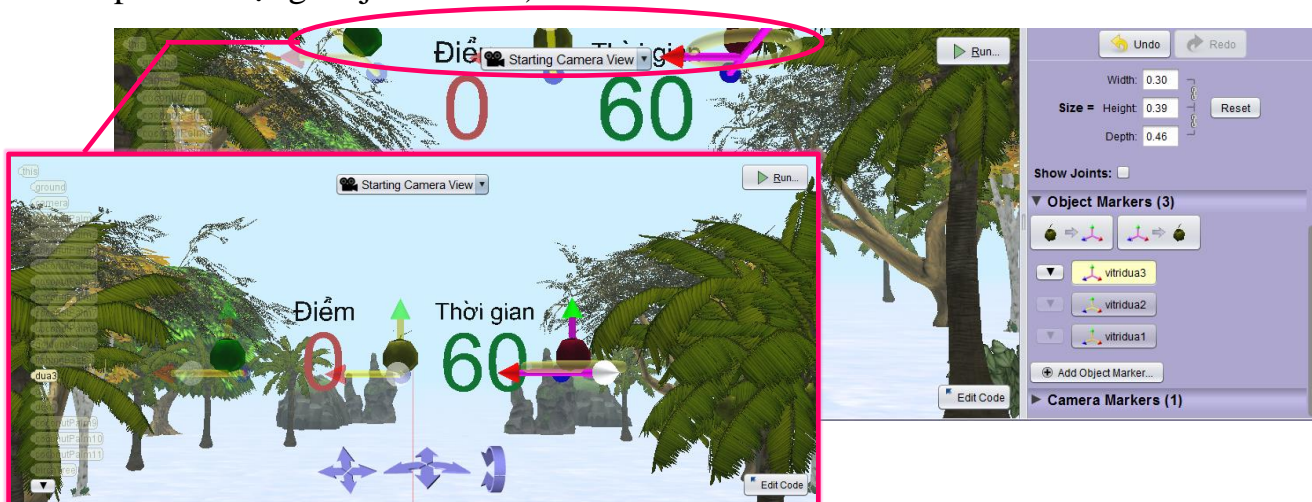
### Lập trình trái dừa rơi (tiếp theo)

Bên trong project Hung\_dua đã kéo sẵn 3 trái dừa (dua1, dua2, dua3) (Hình 1)



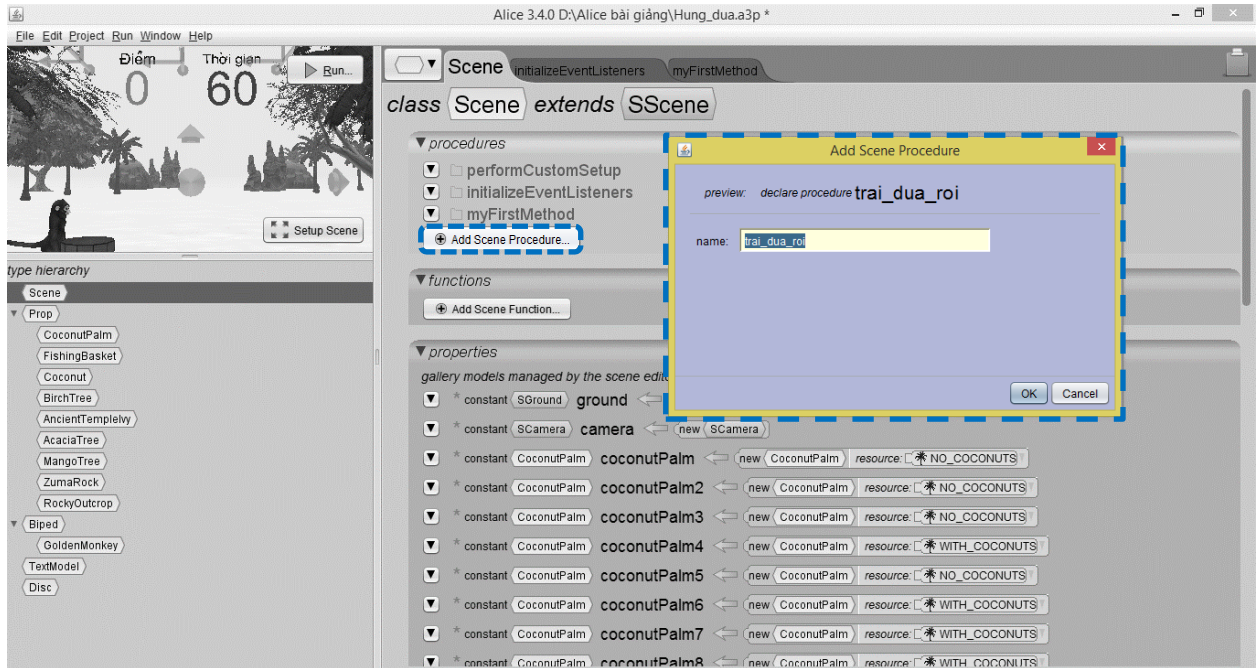
Hình 1: Tên 3 trái dừa đã kéo vào màn hình 3D

Các bạn không thấy được nó vì tôi đã ẩn nó đi. Vị trí của 3 trái dừa sẽ trùng với vị trí của ObjectMarker (Hình 2). (Ở hoạt động sau sẽ giải thích rõ hơn tại sao phải sử dụng ObjectMarker).



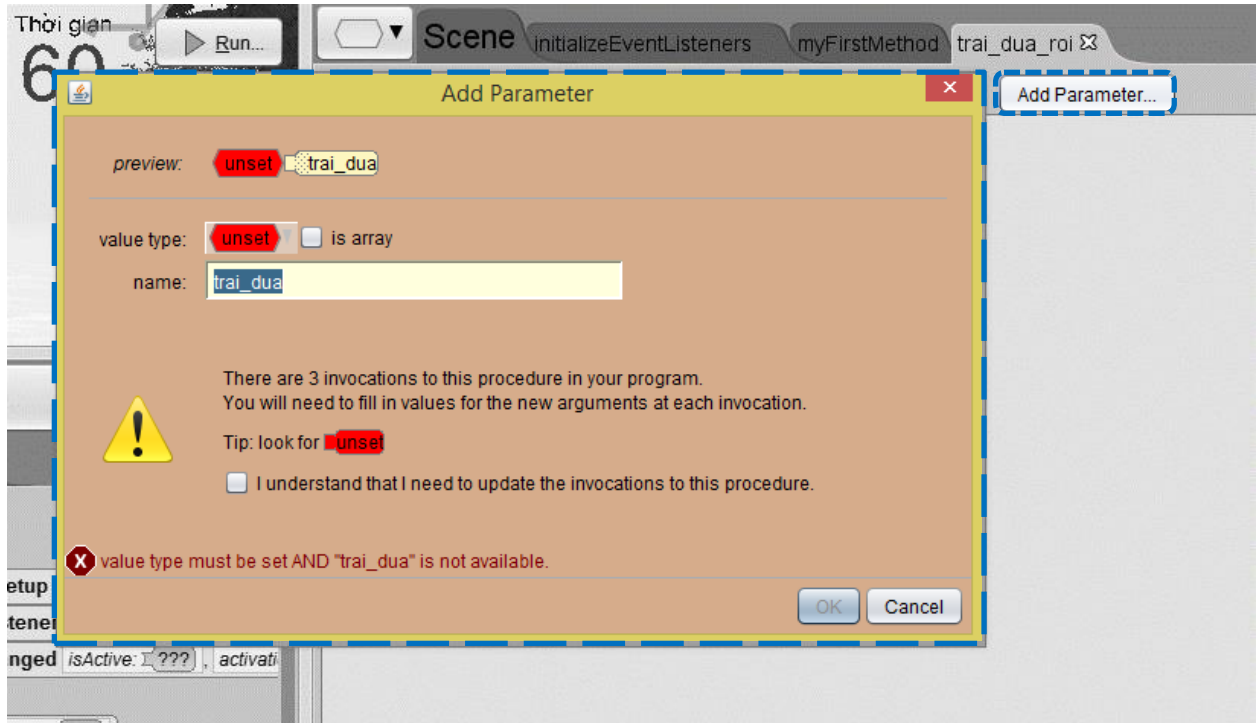
Hình 2: Vị trí của 3 trái dừa khi chưa ẩn đi

Ta sẽ lập trình trái dừa rơi ở riêng một thủ tục. Ta sẽ tạo một thủ tục mới tên “**trai\_dua\_roi**” (Hình 3)



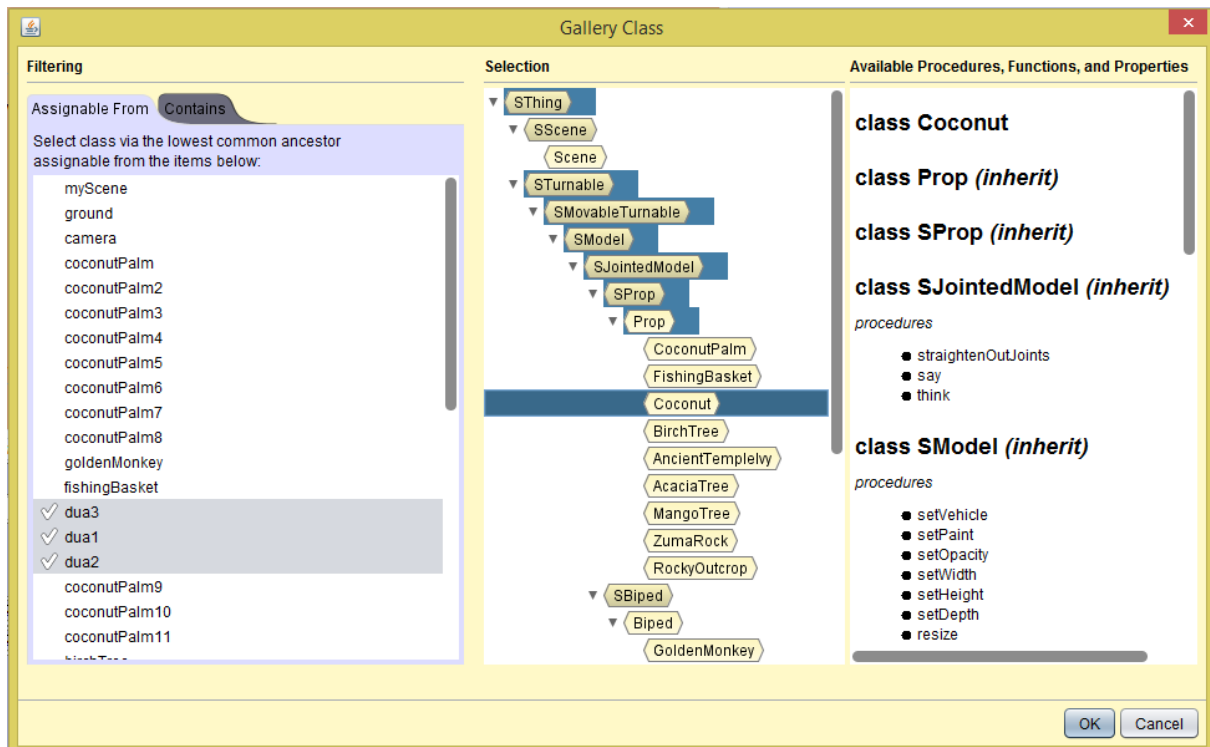
Hình 3: Tạo thủ tục “trai\_dua\_roi”

Ta sẽ thêm một tham số (hay Parameter) có tên là “**trai\_dua**” (Hình 4)



Hình 4: Thêm tham số (Parameter)

Value type ta sẽ chọn là Gallery Class. Xuất hiện bảng chọn Gallery Class, ta chọn Coconut (Hình 5)



Hình 5: Bảng chọn Gallery Class



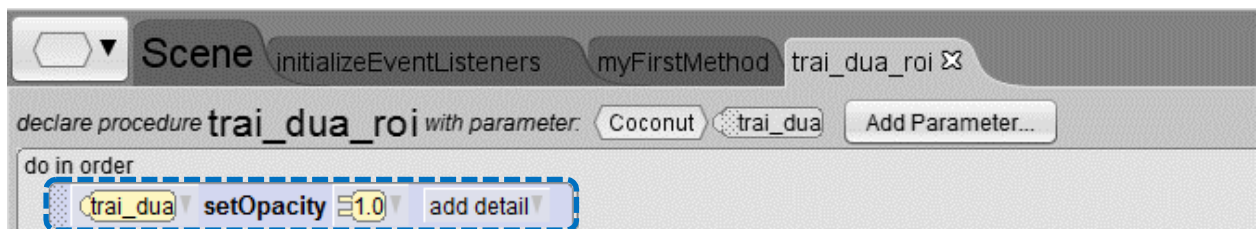
**Tham số (Parameter) dùng để làm gì?** Tham số này là tham số là đại diện cho lớp dừa. Bên trong thủ tục, khi tham số này thực hiện hành động gì thì cũng giống như là lớp dừa đang thực hiện hành động đó. Vì thế, khi gọi thủ tục này cần gọi đến tên trái dừa đã kéo vào màn hình 3D.

Khi tạo xong tham số tham số tạo thành công, tham số **trai\_dua** sẽ xuất hiện trong cây lớp (Hình 6). Ta sẽ chọn nó để lập trình bên trong thủ tục thay cho lớp dừa.



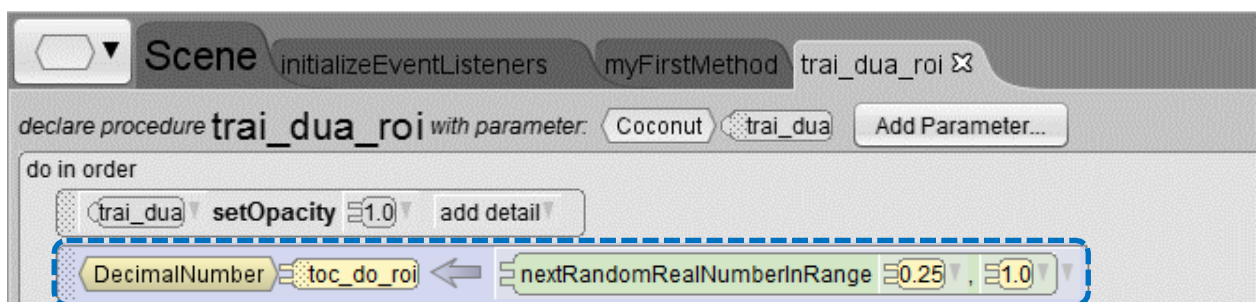
Hình 6: Tham số *traidua*

Đề dứa rơi xuống đất, trước tiên ta phải làm cho dứa xuất hiện. Nên ta sẽ kéo `setOpacity` và chọn độ trong suốt là 1 (Hình 7).

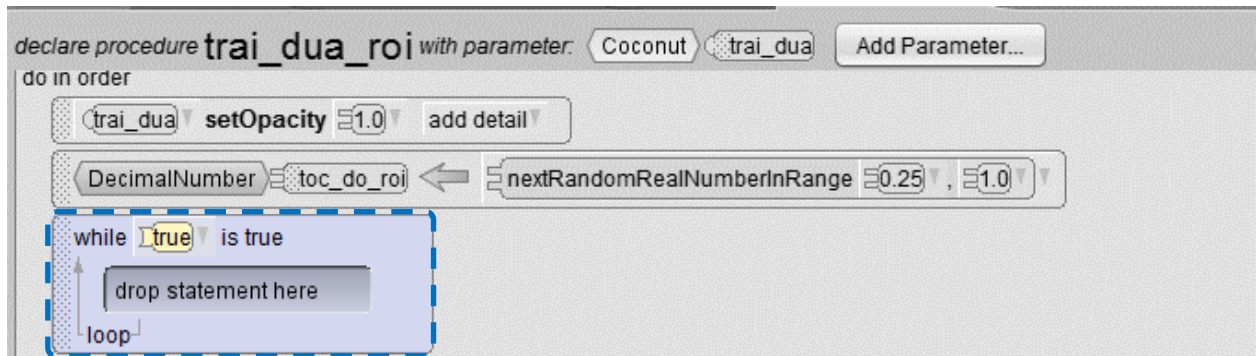


Hình 7: Làm dứa xuất hiện

Ở trò chơi này ta ta thấy trái dứa rơi với tốc độ ngẫu nhiên. Nên ta sẽ tạo một biến “*toc\_do\_roi*”. Biến này sẽ là biến lưu giá trị tốc độ rơi ngẫu nhiên. (Hình 8)

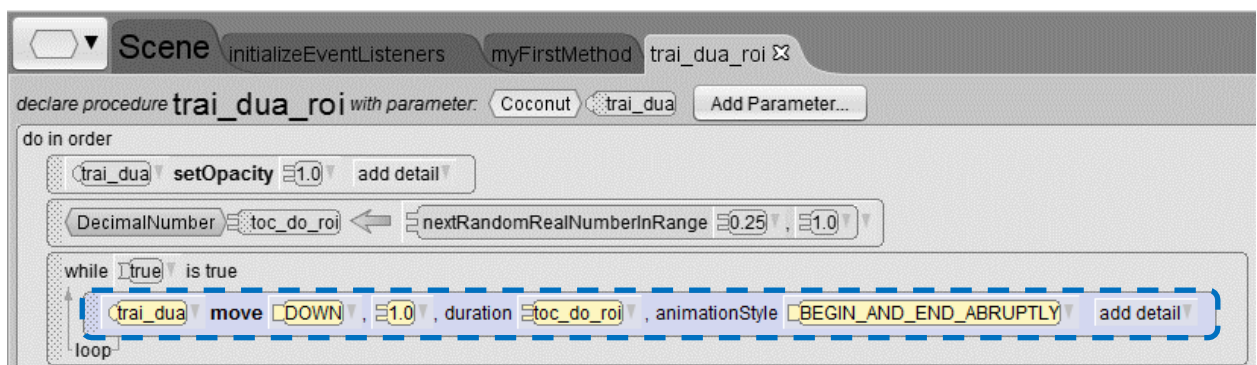
Hình 8: Tạo một biến “*toc\_do\_roi*”

Ta sẽ sử dụng vòng lặp để lập trình cho trái dừa rơi. Ta kéo While vào bảng lập trình . (Hình 9)



Hình 9: Thêm While vào bảng lập trình

Ta sẽ lập trình trái dừa rơi một khoảng bằng 1 với tốc độ là biến **toc\_do\_roi**. Ta sẽ kéo Move của tham số **trai\_dua** vào bên trong while và lập trình các số liệu sao cho phù hợp. (Hình 10)



Hình 10: Lập trình trái dừa rơi

Sau khi đã lập trình cho trái dừa rơi, ta cần điều kiện để dừng trái dừa lại hay xét điều kiện cho vòng lặp While.

### Toán tử logic trong lập trình (Bảng 1)

x	y	x AND y	x OR y	NOT x
Đúng	Đúng	Đúng	Đúng	Sai
Đúng	Sai	Sai	Đúng	Sai
Sai	Đúng	Sai	Đúng	Đúng
Sai	Sai	Sai	Sai	Đúng

Bảng 1: Bảng toán tử logic trong lập trình

*\*Mẹo: VÀ và HOẶC chỉ khác nhau ở 1 điều kiện đúng và 1 điều kiện sai. Nếu tình huống cần chỉ cần thỏa mãn 1 điều kiện thì ta sẽ dùng HOẶC. (Còn VÀ khi sai 1 điều kiện thì sẽ sai luôn)*

Từ ví dụ thực tế tới lập trình:

Ví dụ về **VÀ**

Giả thuyết: Nếu hôm nay trời không mưa **VÀ** tôi hoàn thành xong bài tập, thì tôi sẽ tới nhà bạn chơi.

x
AND
y

Thực tế: Trời hôm nay mưa to, tôi đã hoàn thành xong bài tập nên

x sai vì trời hôm nay mưa
y đúng

tôi **không** tới nhà bạn chơi

**AND:** 1 trong 2 điều kiện đã sai nên không tới nhà bạn

Thường ta sẽ nói vậy: Tuy hôm nay tôi đã hoàn thành xong nhưng trời đổ mưa rất to nên tôi không đến nhà bạn chơi.

Nhưng nếu thay giả thuyết lại thành **HOẶC** thì sẽ khác

Giả thuyết: Nếu hôm nay trời không mưa **HOẶC** tôi hoàn thành xong bài tập, thì tôi sẽ tới nhà bạn chơi.

x
OR
y

thì tôi sẽ tới nhà bạn chơi.

Thực tế: Trời hôm nay mưa to, tôi đã hoàn thành xong bài tập nên

x sai vì trời hôm nay mưa
y đúng

tôi **vẫn** tới nhà bạn chơi

**OR:** 1 trong 2 điều kiện đã đúng nên tới nhà bạn chơi

### Một ví dụ khác về **HOẶC**:

*Giả thuyết:* Nếu bạn mua cây bút máy này **hoặc** bạn mua cây bút chì kia, thì bạn sẽ được tặng thêm một cục gồm.

*Thực tế 1:* Tôi mua ngay cây bút máy, nên tôi được tặng một cục gồm. (Thỏa một điều kiện nên đúng)

*Thực tế 2:* Vì 2 cây bút giá quá mắc nên tôi không mua cây bút. Vì vậy họ cũng chẳng tặng tôi cục gồm nào (Không thỏa điều kiện nào nên sai)

Xét điều kiện cho trái dừa rơi.

Ta sẽ có 2 điều kiện cho trái dừa ngừng rơi

*Điều kiện 1:* khi dừa chạm rỗ thì trái dừa sẽ dừng lại (x)

*Điều kiện 2:* khi dừa chạm đất thì trái dừa sẽ dừng lại (y)

Ta bắt đầu xét:

Khi ta sử dụng **HOẶC**: Nếu dừa rơi chạm rỗ **hoặc** dừa rơi chạm đất thì dừa dừng lại

**Phân tích:** Giả sử dừa rơi chạm đất: Dừa rơi chạm đất nhưng  
Đúng

dừa không chạm rỗ.  
Sai

Vì **HOẶC** nên 1 điều kiện đúng thì vẫn đúng nên dừa sẽ dừng lại

⇒ Dùng **HOẶC** ngay trường hợp dừa ngừng rơi hoàn toàn đúng

Tuy nhiên, bài toán này ta sử dụng vòng lặp là **while**. Thì ta đã học **While** chỉ thực hiện các lệnh bên trong khi điều kiện đúng. Ta phải tìm điều kiện cho dừa rơi. Ta sẽ có 2 điều kiện cho trái dừa rơi:

*Điều kiện 1:* Dừa rơi khi dừa không chạm đất

*Điều kiện 2:* Dừa rơi khi dừa không chạm rỗ

Ta bắt đầu xét:

Khi ta sử dụng **HOẶC**: Nếu dừa không rơi chạm đất **hoặc** dừa không rơi chạm rỗ thì dừa vẫn tiếp tục rơi



**Phân tích:** Giả sử dừa rơi chạm đất: Dừa rơi chạm đất nhưng

Sai

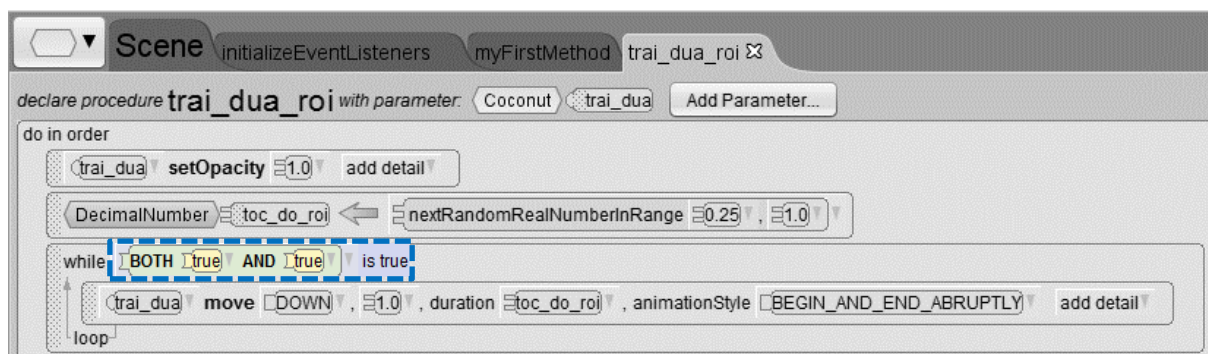
dừa không rơi chạm rỗ.

Đúng

Vì **HOẶC** nên 1 điều kiện đúng thì dừa sẽ vẫn tiếp tục rơi mà **không dừng lại** khi chạm đất.

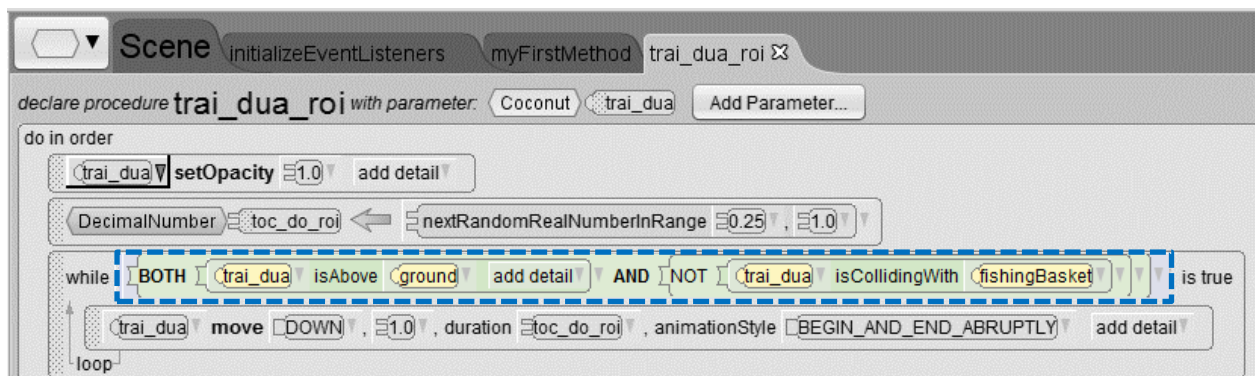
⇒ Vì vậy ta thấy khi sử dụng **HOẶC**, dừa rơi **không chính xác**

⇒ Điều kiện của While ta phải sử dụng **VÀ**. Để trái dừa tiếp tục rơi thì dừa phải ở phía trên đất **VÀ** dừa không được chạm rỗ. Nên điều kiện của while ta sẽ dùng **BOTH...AND** (Hình 11)



Hình 11: BOTH...AND

Sau đó kéo hàm ở phía trên mặt đất (is Above ground) và thêm NOT ở phía trước hàm chạm vào rỗ (NOT... is CollidingWith fishingBasket) (Hình 12)



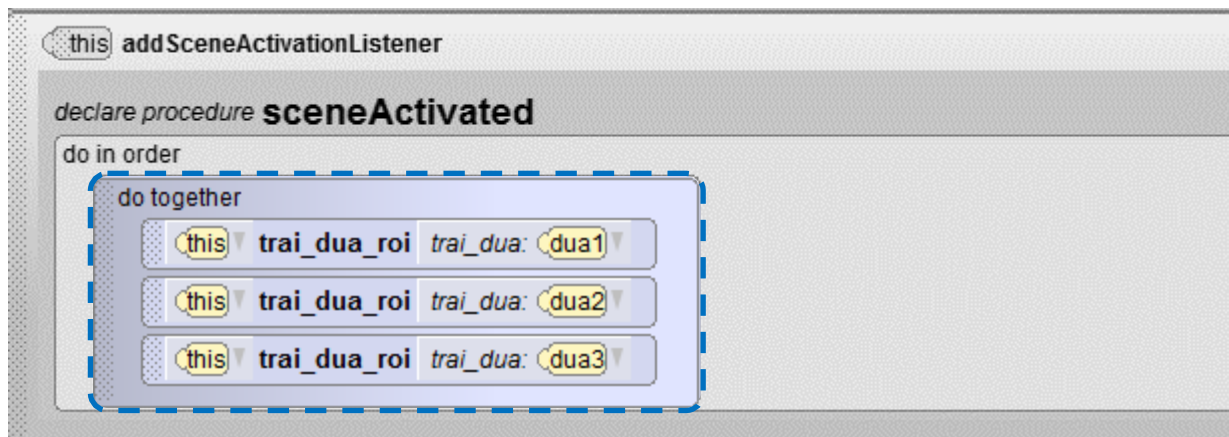
Hình 12: Lập trình điều kiện cho While

Sau đó vào thủ tục initializeEventListener. Ta thêm một sự kiện mới (Hình 13)



Hình 13: Thêm sự kiện Khởi động sự kiện

Sau đó thêm thủ tục `trai_dua_roi` cho 3 trái dừa vào. Để 3 trái dừa rơi cùng lúc thì ta sẽ đặt thủ tục của 3 trái dừa bên trong `do together` (Hình 14)



Hình 14: Đưa thủ tục vào trong sự kiện

-Hết-