Giới thiệu về Game đua xe

(Game đua xe đơn giản với góc nhìn từ trên xuống)  
Phần này sẽ giúp bạn biết cách tạo các đối tượng di chuyển với sự thay đổi hướng di chuyển, vận tốc, quán tính và ma sát trên địa hình.

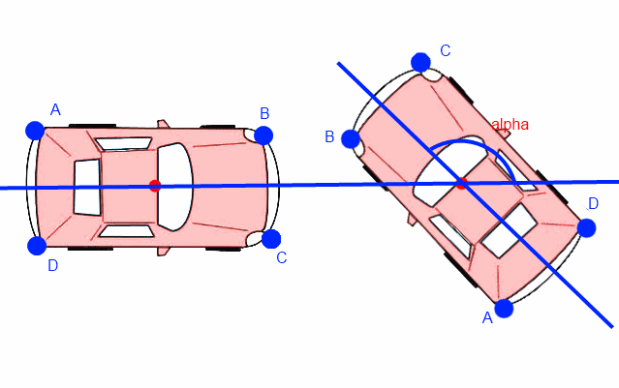
**Các thông số của xe**

Trong game này, ta sẽ sử dụng bốn phím mũi tên để điều khiển xe. Hai phím UP, DOWN để di chuyển tiến lùi và LEFT, RIGHT sẽ dùng để quay hướng xe.  
Một xe đua cơ bản cần các thuộc tính sau:  
– max/min speed: tốc độ tối đa/tối thiểu của xe. Nên để tốc độ tối đa có giá trị tuyệt đối cao hơn tốc độ tối thiểu.  
– acceleration: Khả năng tăng tốc của xe. Giá trị càng lớn thì xe càng nhanh đạt vận tốc tối đa.  
– rotationAngle: Khả năng điều chỉnh góc quay của xe.  
– friction: độ ma sát của xe trên từng loại địa hình. Giá trị này sẽ giúp xe giảm tốc độ khi người chơi không giữ phím di chuyển.

# Di chuyển và quay xe

Với mục đích kiểm tra va chạm, ta sẽ sử dụng một mảng các điểm bao quanh xe hay xảy ra tiếp xúc nhất. Ở đây, các điểm mà tôi chọn là 4 đỉnh từ vùng bao hình chữ nhật của xe (có thể coi như 4 bánh xe). Khi xe xoay góc alpha và di chuyển, ta phải tính lại tọa độ các điểm này tương ứng. Công thức để tính tọa độ mới của một điểm sau khi xoay góc alpha là:

x’ = cos(alpha) \* x − sin(alpha) \* y  
y’ = sin(alpha) \* x + cos(alpha) \* y



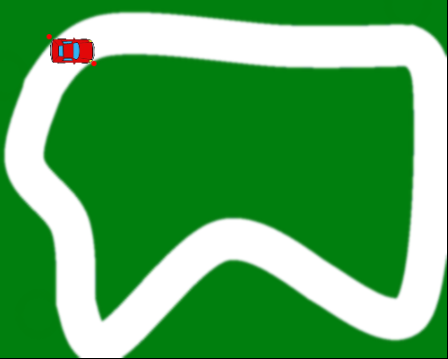
Ví dụ tại góc 0 độ, với gốc tọa độ nằm ở tâm hình chữ nhật, điểm A có tọa độ là {x: -width/2,y: -height/2}. Vậy với góc xoay alpha, tọa độ mới của A là:

x: cos(alpha)\*(-width/2) – sin(alpha)\*(-height/2)  
y: sin(alpha)\*(-width/2) + cos(alpha)\*(-height/2)

# Kiểm tra va chạm (tiếp xúc) với địa hình

Với bản đồ là một hình ảnh, ta cần kiểm tra bằng cách dựa vào pixel. Tuy nhiên không phải bằng cách lặp mà dựa vào kiểm tra một vài vị trí xác định. Các vị trí này ta chính là 4 điểm thuộc các góc của xe mà ta đã xác định trong phần trước.  
– Đầu tiên ta cần lấy đối tượng ImageData của ảnh làm bản đồ.  
– Với mỗi điểm dùng để kiểm tra va chạm của xe, ta tính vị trí alpha của chúng trong ImageData ((x+y\*width)\*4+3) và so sánh giá trị alpha với 0 (tương ứng với mặt đường). Thay vì kiểm tra giá trị alpha, ta có thể kiểm tra màu sắc tại pixel đó cho từng loại địa hình khác nhau.  
– Tăng, giảm ma sát ứng với từng loại địa hình.

Trong ví dụ này, tôi chỉ tạo bản đồ với hai loại địa hình là đường và bãi cỏ. Độ ma sát sẽ tăng dần theo số lượng điểm (bánh xe) tiếp xúc với cỏ:



car.js:

[+ expand source](https://yinyangit.wordpress.com/2012/05/02/html5-canvas-game-dua-xe-part1/?utm_source=zalo&utm_medium=zalo&utm_campaign=zalo)

test.js:

[+ expand source](https://yinyangit.wordpress.com/2012/05/02/html5-canvas-game-dua-xe-part1/?utm_source=zalo&utm_medium=zalo&utm_campaign=zalo)

YinYangIt’s Blog