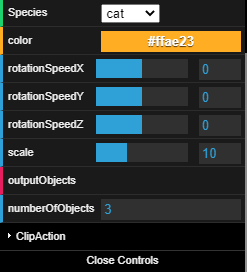
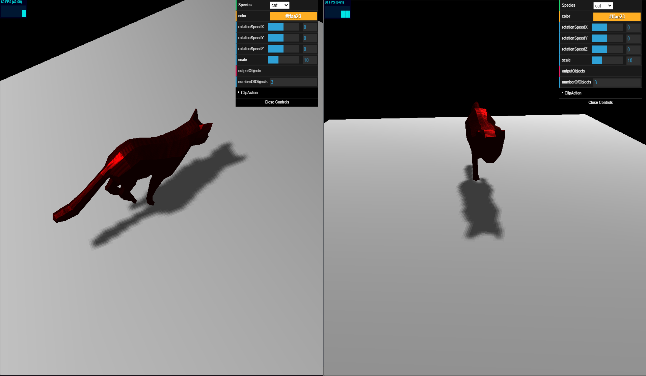
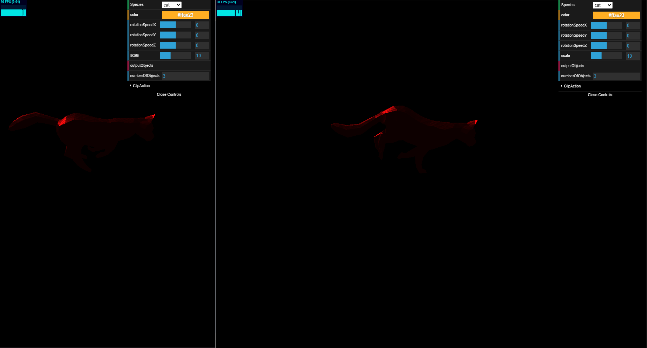
# Program overview

 Ô này hiện tình trạng chạy của web nhờ vào **Stats.js**

Đây là GUI giúp người dùng thay đổi một số cài đặt của mô hình như hướng quay, cỡ, màu …. nhờ vào **dat.gui.js**



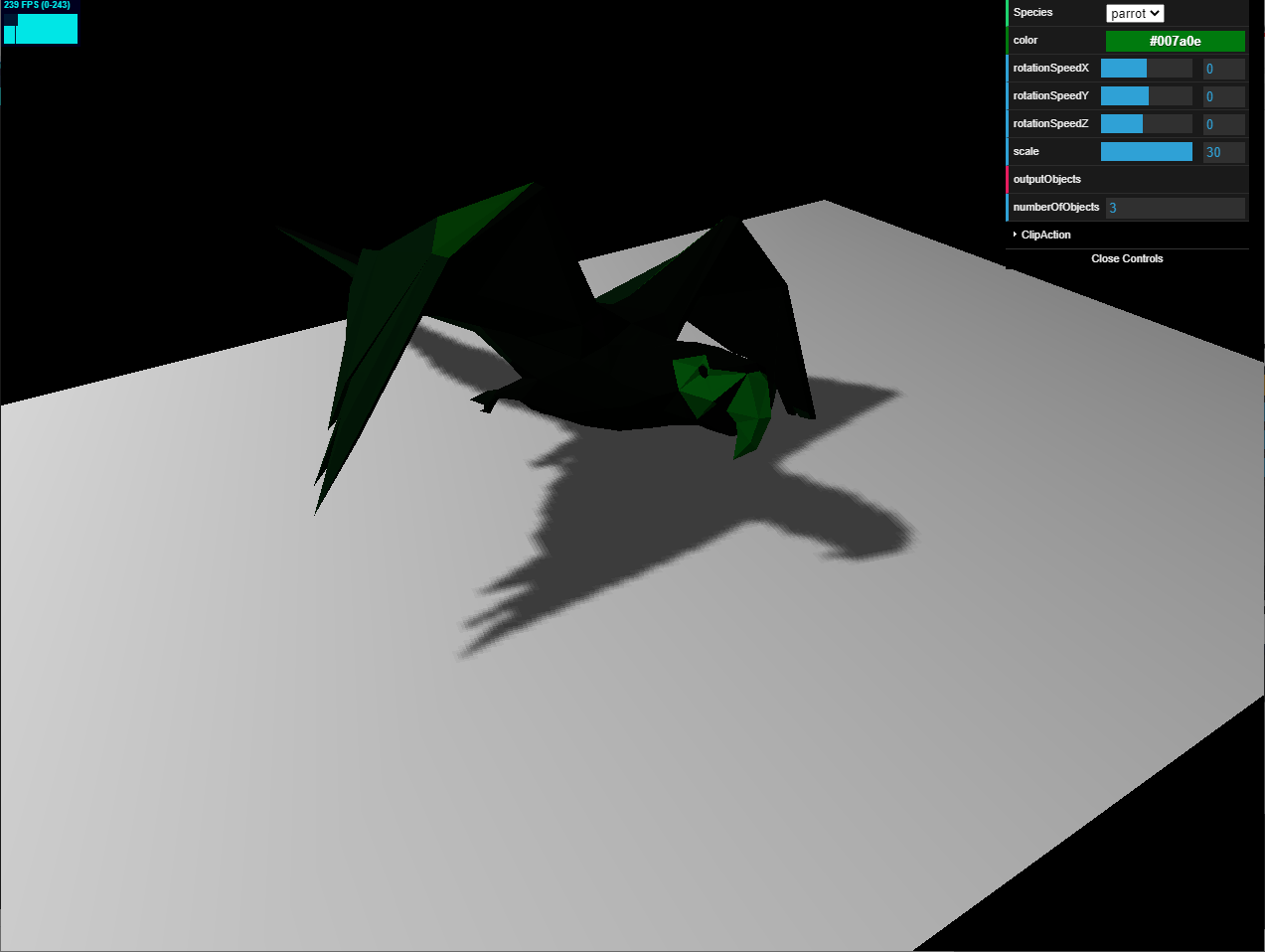
Người dùng có thể sử dụng chuột để di chuyển **camera** nhờ vào **OrbitControls** sau khi đã chuyền cho nó **camera** và **renderer.domElement.**

Người dùng cũng có thể thay đổi kích cỡ màn hình và web sẽ cập nhật lại theo màn hình mới qua đoạn mã **window.addEventListener('resize', onWindowResize, false);**

**'resize'** là một trong các cài đặt của **window** còn **onWindowResize** là tên lời gọi hàm để bảo **camera** và **render** cập nhật với màn hình mới

Khi chạy web, nó đầy tiên sẽ đi qua giai đoạn “chuẩn bị” như tạo renderer, camera, light, orbitcontrol, some Event Listener, panel, gui ... và do em không tìm được một bộ các mô hình, bump map, texture, normal map… nên nó mỗi mô hình sẽ có các cài đặt khác nhau và rất khó để có một đoạn mã chung nên để thuận tiện cho việc tìm hiểu three js, em chia thành 2 chương trình khác nhau là Exam1 để load model và xem một số cài đặt liên quan đến animation và Exam2 model creating em tạo thử một mô hình và các cài đặt liên quan đến nó như vật liệu, vị trí trong không gian.

# Exam1: model loading overview



# Cách chương trình hoạt động

Sau quá trình “chuẩn bị” như chức năng xóa thư mục, tên đường dẫn, gui, ánh sáng, panel…. chương trình sẽ load model có tên là cat đi kèm là tạo ClipAction để có thể thay đổi một số cài đặt của animation.

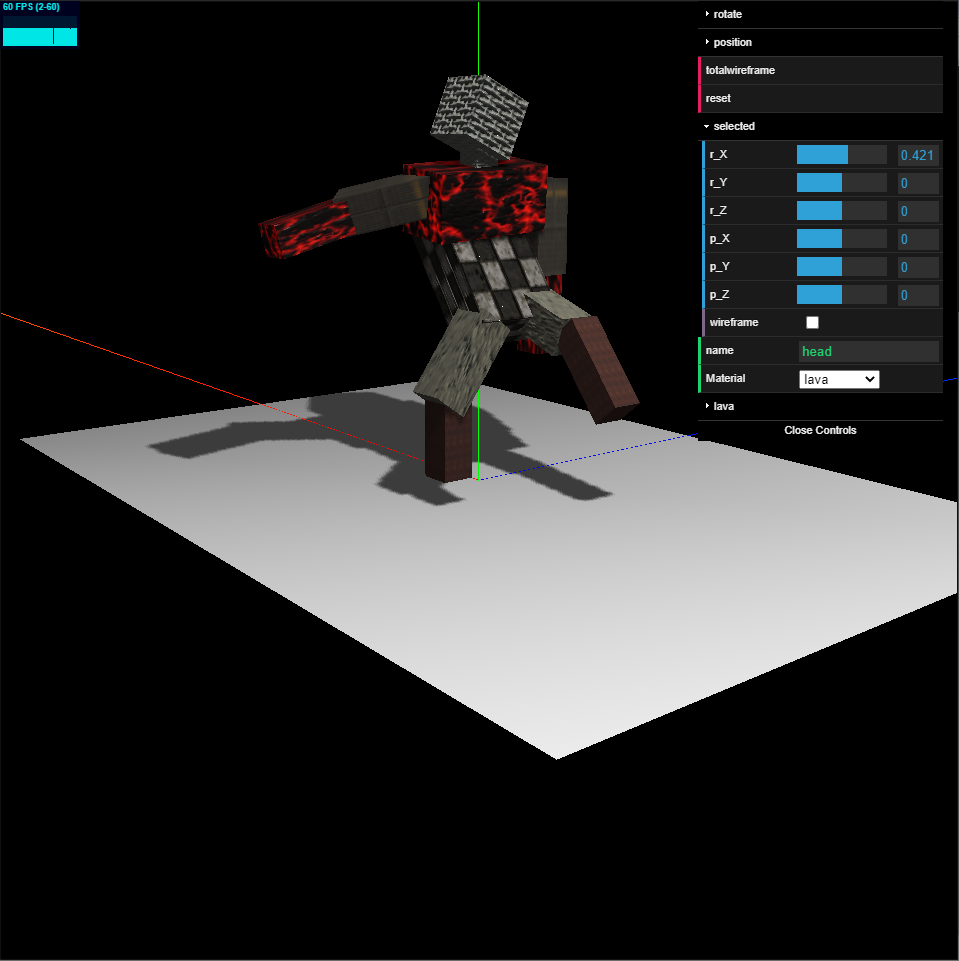
Khi người dùng chọn model khác thì chương trình sẽ xóa mesh và thư mục ClipAction đang có của gui rồi sẽ load model mới, ClipAction mới

Có thể thay đổi màu do gui có thêm đoạn **.listen().onChange(function (e)…)**

**.listen()** để thay đổi giá trị gui theo giá trị đã đổi

**.onChange(function (e)…)** để khi người dùng thay đổi cài đặt của nó thì nó chạy chạy function với e là giá trị được chọn

# Exam2 model creating



# Cách chương trình hoạt động

Hoạt động tương tự như exam 1 nhưng thay vì load model thì chương trình này tạo ra các mesh BoxGeometry với cài đặt vật liệu là gạch ( material như đôi chân trong hình), vị trí, tên.

Người dùng có thể xoay cả mô hình ở mục rotate, di chuyển cả mô hình ở mục position, nút reset giúp mô hình quay lại trạng thái ban đầu và **wireframe** để xem mô hình ở chế độ **wireframe** (nhưng nếu mô hình đấy đang được chọn là **wireframe** rồi thì nó tắt chế độ đấy đi, sẽ tạo nút riêng sau).

Nó hoạt động bằng cách nhét tất cả mesh liên quan đến mô hình này vào một **THREE.Object3D().** Để có thể dễ dàng chọn cả mô hình



Khi có chuột đi qua bộ phận nó sẽ làm vật liệu mờ đi, và người dùng có thể bấm E để chọn bộ phận đó. Điều này sẽ làm vật liệu vẫn mờ kể cả khi di chuột sang bộ phận khác (dự định sẽ thay đôi cách thông báo cho người dùng bời vì nó rất khó nhìn)

Sau đó người dùng có thể cài đặt một số thứ liên quan đến mesh như vật liệu, vị trí của nó

Vật liệu có thể mờ đi do khi bấm E, sẽ chạy function **onDocumentKeyDown** để lưu nó vào biến để khi render sẽ kiểm tra xem có tồn tại biến đấy không, nếu có thì làm mờ

Còn vật liệu mờ khi có chuột nhờ function **onDocumentMouseMove** . Cả 2 function trên đều được thêm vào document qua **.addEventListener**

Ngoài ra khi chọn vật liệu mới cho mesh được chọn thì em tạm sử lí bằng if để tạo gui phù hợp cho từng loại vật liệu

Màu được thay đổi như exam 1