CÁCH LÊN KẾ HOẠCH TỐI ƯU HÓA BẰNG UNITY

**Tóm tắt nội dung**

Unity cung cấp một số công cụ và thiết lập để giúp làm cho trò chơi thực hiện trơn tru. Đối với dự án này, chúng tôi đã chọn những gì chúng tôi nghĩ có thể gây ra nhiều những vấn đề và phân tích cách chúng bị ảnh hưởng tới hiệu suất trò chơi trên bộ xử lý đồ họa Intel®.

Chúng tôi đặt mình vào vị trí của một nhà phát triển Game học hỏi làm thế nào để sử dụng Unity. Chúng tôi muốn va vấp vào những khó khăn về hiệu suất và sau đó xác định làm thế nào để làm việc thông qua các vấn đề với cơ chế hoạt động xây dựng trong Unity. Một trong những điểm mạnh của Unity là khả năng tạo ra các nội dung một cách nhanh chóng, nhưng khi xem xét hiệu suất, đặc biệt là trên các thiết bị di động và máy tính bảng, các nhà phát triển cần phải giảm tốc độ và lên kế hoạch làm thế nào để sử dụng được bản built trong cơ chế thực hiện. Bài viết này chuẩn bị cho người dùng Unity cũ và mới với các cân nhắc thực hiện khi xây dựng levers/game của bạn, và cung cấp những cách thức mới để xây dựng

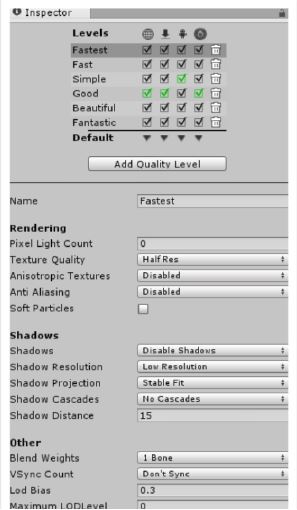
**Giới thiệu**

Tạo trò chơi trong Unity là khá đơn giản. Unity cung cấp một store nơi bạn có thể mua các sản phẩm như mesh, script viết sẵn, demo game hoặc ngay cả một trò chơi hoàn thiện. Vì mục đích thử nghiệm của tôi, tôi chỉ quan tâm đến các thao tác của một Game hiện có để tìm khu vực thể hoặc không thể đạt được những lợi ích hiệu năng. Tôi nghiên cứu sâu về Unity Tech Demo gọi là Boot Camp, tải miễn phí tại assets store, để xem những vấn đề tôi có thể gặp phải.

Tôi sử dụng Unity 3.0 để tạo ra các game settings và chạy tất cả các scene. Các thử nghiệm đã được thực hiện trên một máy tính thế hệ thứ 3 Intel® Core ™ xử lý trên Intel® HD Graphics 4000. Các kết quả kiểm tra không được áp dụng với các thiết bị di động.

**Quản lý chất lượng**

Unity đã thêm render settings cho các trò chơi tìm thấy trong: Edit->Project Settings->Quality menu (Figure 1). Chúng có thể tùy chỉnh render settings để có thể phục vụ cho nhu cầu cá nhân. Unity có tài liệu trực tuyến hữu ích để giải thích những gì liên quan đến Quality Settings và làm thế nào để thay đổi các thiết lập thông qua kịch bản API của Unity .



*Figure 1*

Còn về nhiệm vụ của tôi để tối ưu hóa với Unity, tôi quyết định để mess xung quanh với một số các thiết lập chất lượng để xem những khó khăn và thuận lợi có thể tìm thấy, mặc dù tôi đã không kiểm tra tất cả các tùy chọn khác nhau có sẵn.

Texture Quality (Chất lượng Texture)

Cài đặt Quality Inspector có một Drop Down Menu để bạn lựa độ phân giải cho các texture của bạn. Bạn có thể chọn từ 1/8, ¼, ½ hoặc Full Resolution. Để xem các lợi ích/bất lợi về hiệu năng giữa các texture có độ phân giải khác nhau, chúng ta lấy bản chụp tốc độ khung hình của một cảnh mẫu, kiểm tra tất cả các Quality Settings mặc định của Unity(Fastest, Fast, Good, etc.).

Việc tối ưu Quality Texture chủ yếu dựa vào mắt nhìn của người dùng theo 2 tiêu chí : Đảm bảo chất lượng hình ảnh người dùng và tỉ lệ phải thích hợp (không lớn quá nhưng cũng nên nhỏ quá) so với Full Resolusion để làm tăng hiệu suất

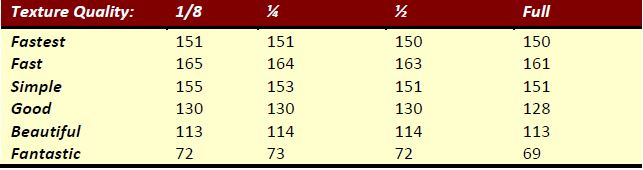


*Ảnh chụp với Quality Texture 1/8 Full Resolusion*



*Ảnh chụp với Quality Texture Full Resolusion*

Với 2 ảnh chụp ta thấy rất rõ sự khác biệt giữa việc để Full Resolusion và 1/8 Full Resolision . Tuy nhiên việc để chất lượng thấp nhất 1/8 Full Resolusion chưa chắc đã làm tăng hiệu suất tối đa của Game :



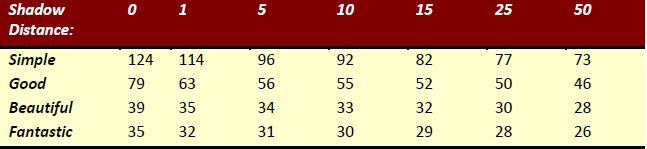
*Bảng đo FPS của một Scene mẫu dựa vào Texture Quality ở các Levers khác nhau*

Khoảng cách bóng (Shadow Distance)

Shadow Distance là một thiết lập để thay đổi khoảng cách triệt tiêu bóng của đối tượng trong môi trường có ánh sáng chiếu từ 1 phía . Khoảng cách này được tính bằng khoảng cách từ nguồn sáng (Light) đến đối tượng . Ví dụ: Ta đặt Shadow Distance là 100 thì đối tượng cách Camera <= 100m thì sẽ nhìn thấy bóng của đối tượng ; >100m thì không thấy bóng của đối tượng . Lưu ý khi đã khoảng cách đã đạt yêu cầu để nhìn thấy bóng của đối tượng thì khoảng cách càng gần bóng của đối tượng càng rõ , tạo nên độ chân thực và chiều sâu cho đối tượng



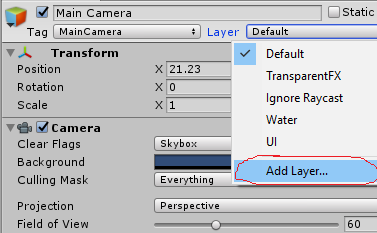
Bảng đo FPS với các mức Shadow Distance khác nhau ở các Levers khác nhau

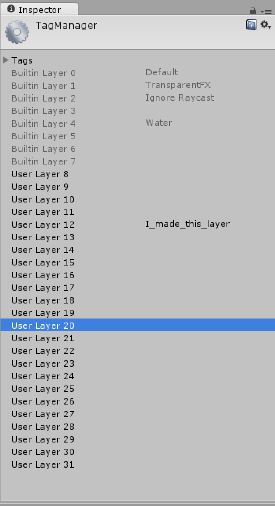


Nhìn chung chúng ta đề thấy hiệu năng của game càng giảm khi Shadow Distance càng cao , vì vậy cần thiết lập một cách hợp lý

**Layer**

Tất cả các đối tượng trò chơi bên trong Unity được gán cho một layer khi tạo . Các đối tượng khi tạo đều được đặt layer Default . Chúng ta có thể tạo ra một lớp bằng cách 2 cách : Ấn vào Default ở Layer và Add new hoặc Edit->Project Settings->Tags



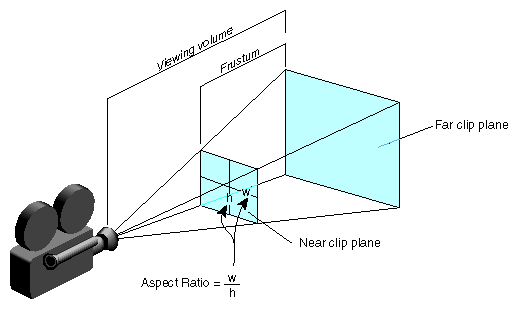


*Cửa sổ quản lý Layer*

Cả 2 cách đều hiện ra cửa sổ quản lý như hình trên . Chúng ta có thể tạo ra một layer mới và xác định số của lớp . Layer mới tạo ra có thể được dùng để nhóm các đối tượng lại với nhau để dễ sử dụng và thao tác

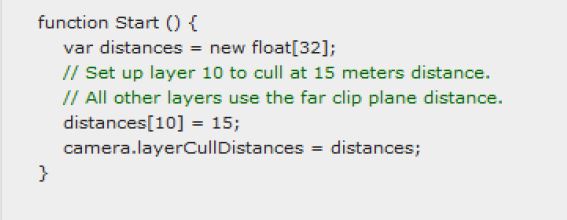
Layer Cull Distances

Các đối tượng nằm ngoài mặt phẳng cắt của Camera thì sẽ không được camera nhìn thấy .



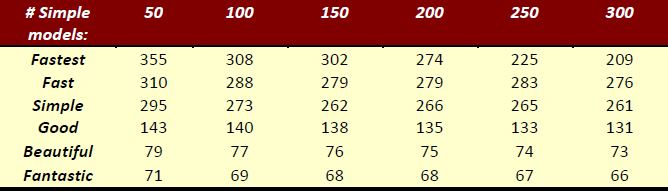
*Hai mặt phẳng cắt gấn và xa của Camera*

Nhưng có một cách , thông qua một Script , đặt vào một mặt phẳng cắt có kích thước thích hợp hơn

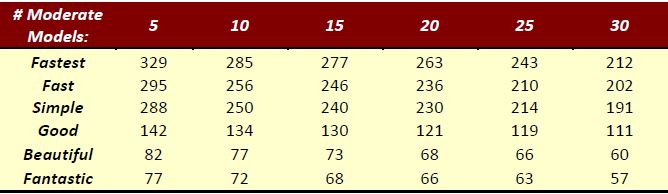


Trong đoạn code layer 9 (ứng với distance[10]) được gán độ xa có thể nhìn thấy là 15m. Vì vậy tất cả các đối tượng nào được gắn Layer này có thể được camera nhìn thấy trong khoảng cách nhỏ hơn 15m kể cả độ xa có thể nhìn thấy của Camera có thể lớn hơn 15m

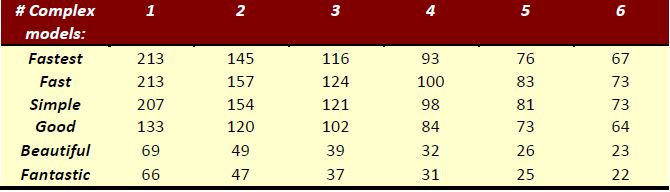
Bảng thống kê chi tiết :



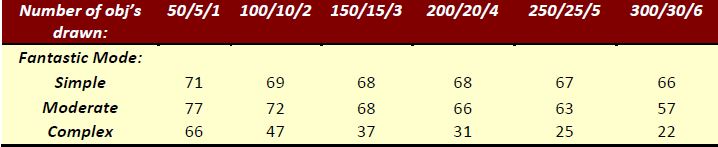
*Models đơn giản*



*Models vừa phải*



*Models phức tạp*

**

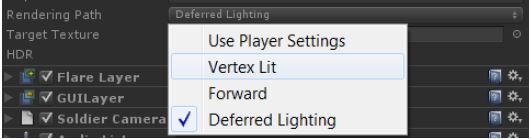
*Tổng hợp các kết quả trên*

**Camera**

Với camera chúng ta tập chung vào phần cài đặt các tính năng của nó như : Rendering Path và HDR.

Render Path

Dùng để xử lý ánh sáng và bóng tối trong Game . Các lựa chọn sẽ sắp xếp theo độ phức tạp tăng dần , các lựa chọn này còn dựa vào sức mạnh xử lý của CPU và GPU . Nếu sự lựa chọn của người dùng mà CPU và GPU không đáp ứng được thì Unity sẽ tự động chuyển sang chế độ ít phức tạp hơn để xử lý



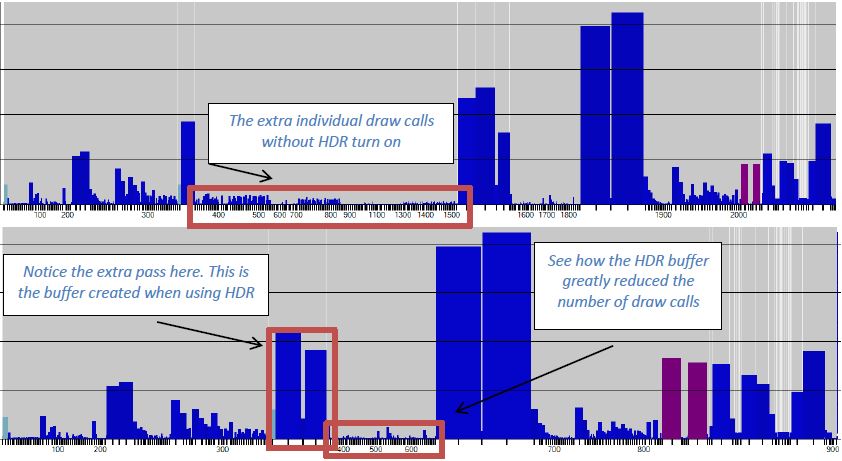
Các nhà phát triển Unity cần cập nhật dựa vào cấu hình thiết bị của người dùng để chọn những mức xử lý thích hợp

HDR

Trong Renderring bình thường , mỗi điểm ảnh của giá trị màu đỏ , xanh lá , xanh dương được đại diện bởi một số thập phân trong khoảng [0, 1] chúng ta có thể thay đổi các giá trị đó để ánh sáng có thể nhìn tự nhiên và chân thực hơn , Unity gọi đó là HDR Setting . HDR cho phép chúng ta thay đổi giá trị đại diện của các màu trên vượt ra khỏi mức bình thường của chúng , có thể là nằm ngoài khoảng [0, 1] . HDR tạo ra một vùng đệm hình ảnh có hỗ trợ các giá trị bên ngoài phạm vi của 0-1, và thực hiện các hiệu ứng hình ảnh sau xử lý . Sau khi xử hình ảnh, các giá trị R, G và B trong bộ đệm hình ảnh mới được tạo ra được thiết lập lại các giá trị trong phạm vi 0-1 bằng các hình ảnh Unity Effect Tonemapping

Nếu xử dụng Forward Rendering , HDR sẽ chỉ hoạt động khi các hiệu ứng hình ảnh xuất hiện

Nếu xử dụng Deferred Rendering và có hiệu ứng hình ảnh gắn liền với một Camera, HDR nên được kích hoạt



*So sánh Draw calls của một Scene mẫu với Image Effect và Deferred*

**Image Effects**

Unity Pro đi kèm với một loạt các hiệu ứng hình ảnh tăng cường giao diện của một cảnh . Thêm Image Effects, thậm chí sau khi tạo dự án của bạn, bằng cách vào Assets-> Import Package-> Image Effects. Sau khi nhập, có hai cách để thêm hiệu ứng cho ảnh. Nhấp chuột vào đối tượng máy ảnh trò chơi của bạn, sau đó trong giao diện máy ảnh, chọn Add Component, sau đó chọn Image Effects . Cũng có thể nhấp chuột vào đối tượng máy ảnh từ hệ thống menu bằng cách vào Component-> Image Effect.

SSAO – Screen Space Ambient Occlusion

SSAO là một hiệu ứng hình ảnh bao gồm trong gói Unity Pro’s Image Effect. Hình dưới cho thấy sự khác biệt giữa một cảnh với SSAO khi bật và tắt. Các hình ảnh trông tương tự, nhưng hiệu suất là khác nhau rõ rệt. Các cảnh mà không có SSAO chạy ở 32 FPS và cảnh với SSAO chạy ở 24 FPS, giảm 25%.



Nên cẩn thận khi thêm hiệu ứng hình ảnh, vì họ có thể ảnh hưởng tiêu cực đến hiệu suất

Occlusion Culling

Occlusion Culling làm vô hiệu hóa những đối tượng nằm sau mặt cắt của Camera và kể cả các đối tượng ẩn sau các đối tượng khác . Điều này làm cho số lượng thông tin phải xử lý giảm đi và rất có lợi cho hiệu xuất . Tuy nhiên việc thiết lập nguyên tắc tiêu hủy đối tượng là khá phức tạp . Khi thiết lập nguyên tắc này cần tìm hiểu 2 thuật ngữ :

* Occluder : Một đối tượng được đánh dấu là một occluder hoạt động như một rào cản ngăn chặn đối tượng khác ; đánh dấu là occludees không bị trả lại.
* Occludee : Đánh dấu một đối tượng trò chơi như một occludee sẽ quy định đối tượng đó sẽ không bị ngăn chặn bởi các Occluder



Hai bức ảnh trên là hiệu xuất khi xử dụng và không sử dụng Occlusion Culling . Cả 2 Scene mẫu , đằng sau bức ảnh là một loạt các đối tượng phức tạp xử dụng Mesh Collider . Khi xử dụng Occlusion Culling thì hiệu xuất tăng lên khá đáng kể.

Level of Detail (LOD)

LOD cho phép nhiều Mesh được gắn vào một đối tượng để chuyển đổi các Mesh đó dựa trên khoảng cách của Camera . Nó mang lại lợi ích cho các đối tượng ở xa Camera . Nó có thể đơn giản hóa các Mesh để cho hiệu xuất tốt nhất .



Ở trên là kết quả đo được với một Scene mẫu gồm tập hợp một số ngôi nhà , mỗi ngôi nhà được gắn 3 Mesh khác nhau với các lever khác nhau :

* Lever 0 : Best Quality
* Lever 1 : Medium Quality
* Lever 2 : Low Quality

Mặc dù cách sử dụng LOD mang lại rất nhiều lợi ích cho hiệu xuất nhưng việc này lại tốn khá nhiều công sức , đòi hỏi nhà thiết kế 3D phải tạo ra nhiều Mesh khác nhau cho mỗi đối tượng.

Batching

Các đối tượng trên màn hình cũng có thể là nguyên nhân gây ra việc hiệu xuất bị giảm . Các đối tượng để Static và Dynamic cũng đều có những hiệu xuất khác nhau . Một lời khuyên là những đối tượng không di chuyển , cố định thì nên dùng Static để tăng hiệu xuất của chúng





Bảng trên là kết quả đo được khi sử dụng các đối tượng (Máy bay) ở chế độ Static hoặc không .

Kết luận

Unity là một công cụ dễ phát triển nhưng nó cũng có rất nhiều rắc rối về hiệu xuất . Tuy rằng Unity cung cấp cho chúng ta những công cụ để tối ưu hóa hiệu xuất , nhưng không phải tất cả đều trực quan và dễ dàng thiết lập . Có những công cụ mà chỉ cần bật , tắt không hợp lý là có thể ảnh hưởng tiêu cực tới hiệu xuất của Game . Vì vậy trước khi thực hiện tối ưu hóa chúng ta cần tạo kế hoạch cụ thể để việc tối ưu được dễ dàng hơn .